

JPRS 75353

21 March 1980

East Europe Report

SCIENTIFIC AFFAIRS

No. 666



FOREIGN BROADCAST INFORMATION SERVICE

NOTE

JPRS publications contain information primarily from foreign newspapers, periodicals and books, but also from news agency transmissions and broadcasts. Materials from foreign-language sources are translated; those from English-language sources are transcribed or reprinted, with the original phrasing and other characteristics retained.

Headlines, editorial reports, and material enclosed in brackets [] are supplied by JPRS. Processing indicators such as [Text] or [Excerpt] in the first line of each item, or following the last line of a brief, indicate how the original information was processed. Where no processing indicator is given, the information was summarized or extracted.

Unfamiliar names rendered phonetically or transliterated are enclosed in parentheses. Words or names preceded by a question mark and enclosed in parentheses were not clear in the original but have been supplied as appropriate in context. Other unattributed parenthetical notes within the body of an item originate with the source. Times within items are as given by source.

The contents of this publication in no way represent the policies, views or attitudes of the U.S. Government.

PROCUREMENT OF PUBLICATIONS

JPRS publications may be ordered from the National Technical Information Service, Springfield, Virginia 22161. In ordering, it is recommended that the JPRS number, title, date and author, if applicable, of publication be cited.

Current JPRS publications are announced in Government Reports Announcements issued semi-monthly by the National Technical Information Service, and are listed in the Monthly Catalog of U.S. Government Publications issued by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

Indexes to this report (by keyword, author, personal names, title and series) are available from Bell & Howell, Old Mansfield Road, Wooster, Ohio 44691.

Correspondence pertaining to matters other than procurement may be addressed to Joint Publications Research Service, 1000 North Glebe Road, Arlington, Virginia 22201.

March 1980

EAST EUROPE REPORT

SCIENTIFIC AFFAIRS

No. 666

CONTENTS

PAGE

BULGARIA

Project Examines Problems of Pharaoh's Ants (Gospodin Sveshtarov; RABOTNICHESKO DELO, 4 Feb 80)...	1
Robotics-Made Robots Described (Iliana Nencheva, Lyubomir Nikolayev; POGLED, 1 Feb 80)	3
Chemical Research Work at Unified Center for Chemistry (Borislav Kurtev; VECHERNI NOVINI, 16 Feb 80).....	6
Report on New Patents, Technologies, Materials (ELEKTROPROMISHLENOST I PRIBOROSTROENIE, No 11, 1979).	9
New Patents and Inventions	
New Products, Materials	
Activities in Institutes, Enterprises	

HUNGARY

Feasibility of Computers Based on Non-Neumann Principles (Gabor David; INFORMACIO ELEKTRONIKA, No 1, 1980)....	20
Logic Machines Versus Neumann-Principle Computers in Task Solving (Tamas Gergely, Miklos Szots; INFORMACIO ELEKTRONIKA, No 1, 1980)	36

POLAND

Optical Tracking Station, Laser Range-Finder Described (Pawel Elsztein; SKRZYDLATA POLSKA, 25 Nov 79)	51
--	----

BIOLOGIST EXAMINES PROBLEMS OF PHARAOH'S ANTS

Sofia RABOTNICHESKO DELO in Bulgarian 4 Feb 80 p 2

[Article by Gospodin Sveshtarov, biologist: "Uninvited Guest"]

[Text] Biologists know the pharaoh's ant by the name of monobrium pharaonis. This species was described for the first time in 1758 by the noted Swedish naturalist Carolus Linnaeus after he had found it in the mummies of the Egyptian pharaohs. Later on myrmecologists (experts who study biology and ant migration) established that the pharaoh's ant was a tropical species and that it had spread all over the world from its original fatherland, the jungles of India.

In the beginning of the thirties of this century our noted myrmecologist, Dr Neno Atanasov, discovered the presence of this species of ant in one of the Burgas hotels. In 1951 he discovered a colony of pharaoh's ants in the old hotel in Svilengrad, and in 1954 in the hotel in Michurin. In the following 10-15 years the ants started to appear in other districts of the country as well.

At first nobody paid any attention to the presence of these minute and totally nondescript insects. To our regret, however, they have already adapted to such an extent to the environment and the food in people's homes that if let to live in the forest they would probably die fast.

Another reason for their adaptability is their friendliness. The separate nests of pharaoh's ants live together in most friendly and neighborly relations and never engage in wars. And finally, it seems that for those extremely tiny insects there are no obstacles in penetrating the human dwellings. They squeeze through the smallest cracks, move from place to place along the heating, water or sewage pipes, through small holes in the walls, windows, and doors and greedily attack the food in the homes and public buildings. Unlike some of their other brethren, the pharaoh's ants prefer proteins. Therefore, we find them by the thousands on meat, hard boiled eggs, cold cuts, nuts, etc.

Pharaoh's ants build their nests in houses that offer them most advantageous conditions for starting a colony. Such are, for example, dwellings with a heating system, warm and dry storehouses, laboratories, and pantries where cold cuts and sweets are stored. They build their nests most often in the walls, up to 1 meter deep. As soon as they feel comfortably settled down, the minute "thieves" overrun the entire building looking for food. Their workers are strongly attracted by mouth and skin secretions and are, therefore, particularly dangerous for hospitals. There have been cases when they had been found on fresh wounds, under bandages and even on surgical instruments. By such feats the pharaoh's ants are apt to transmit bacterial infections, most often streptococci. On the other hand, they also inflict economic damages, namely, they devour each year a considerable amount of food and contaminate an equal amount of other foodstuffs, which then become unfit to eat.

We still have no effective measures against the spread of pharaoh's ants. Quite often some homeowners discover a nest and destroy all ants. Soon, however, the pleasure of the accomplishment evaporates, as new reinforcements of the annoying insects appear, coming from other nests in the house that have not been located nor affected by the disinfecting agent.

We use, with limited success, a solution of Negovon preparation, which is poisonous and requires great care. Thus, for the time being, the most effective way to protect ourselves from the ants' invasion is total destruction of all nests. For this purpose the citizens should seek the help of the KhET [Hygiene-Epidemiological Institute] specialists, who fight the uninvited guests by scientifically based methods.

1010

CSO: 2202

BULGARIAN-MADE ROBOTS DESCRIBED

Sofia POGLED in Bulgarian 18 Feb 80 p 16

[Article by Iliana Nencheva and Lyubomir Nikolayev: "The Robots Are Coming"]

[Text] Automation and robots have moved from science fiction novels into modern industry and production. But outside of narrow specialized groups, the concept, history and essence of robots remain rather hazy.

Where are robots being employed? The automation of production operations requiring manual labor, the transporting of goods under the conditions of difficult terrain, the identification and analysis of situations--these are certain areas of human activities in which robots are used.

Robots are capable not only of performing many and diverse operations (tasks), but also can be effectively instructed and retrained from one job to another.

What is the history of robots?

The first generation robots were robots with program control. They were designed chiefly to perform a strictly programmed sequence of operations which was determined by the production process.

At present there are around 300 types of programmed robots (around 100 in Japan, 30 in the United States, and the remainder in different European countries). These are used in machine building, metallurgy and the nuclear industry.

The second generation robots are the so-called robots "with feelings." Their difference from the first generation is that they are equipped with a large quantity of diverse sensors and a complicated control system. Due to the new capabilities provided in them to perceive changes in the environment, to analyze sensory information and adapt to the existing operating conditions, these robots can handle unoriented parts of an arbitrary form, they can perform installation work, and collect information on an unknown, changing external environment. They are used in space science and certain other scientific activities, but at present are not in series production.

The third generation robots are the intelligent (thinking) robots. These differ fundamentally from the second generation ones in terms of complexity and refinement of the control system which includes elements of artificial intellect. It must be remembered that the intelligent robots are certainly not designed just to simulate the physical actions of man, rather to automate his mental activity, that is, to solve intellectual problems.

In recent years, as a consequence of the accelerated development of technical cybernetics, in our nation the real prerequisites have been created for the setting up of a new area under electronics and cybernetics, namely robot engineering. The decisions of the July Plenum affirmed the necessity of a further improvement, development and introduction of robots, and defined this as a strategic area, in confronting it with the task of the integrated and profound, further development and introduction of this area into production.

Within CHMA Bulgaria is the coordinator of the robot engineering program. The first serially produced industrial robots are Bulgarian. These have been well received in a number of nations such as the USSR, Poland, the GDR and elsewhere. Exhibits of Bulgarian-produced robots were demonstrated at the Robots-79 Exhibit.

The robots produced in Bulgaria can be divided into several types:

- 1) The pneumatic balanced manual manipulators which are employed in forging-stamping and press production;
- 2) The portable Pirin manipulators which are a family of product-lifting manipulators;
- 3) The standing Pirin manipulators which are produced in four modifications designed for supplying parts to metal-cutting machines;
- 4) The RB-110 industrial robots which are used to automate certain manipulations in machine building and in other areas, for example, in the production of plastics, glass and ceramics;
- 5) The RB-210 industrial robots which are used for applying powdered coverings, but it is also possible to use them in spraying;
- 6) The RB-230 industrial robots which are used for automating basic and auxiliary production operations.

Where are the programs being worked out? The senior group specialized in the questions of robot engineering in our nation is the Scientific Research Center for Robot Engineering under the V. I. Lenin Higher Electrical Machinery Institute under the leadership of Professor and Doctor of Technical Sciences Yuliyán Marinov. The two main areas where the center has begun to work are scientific research activities and the training of junior specialists. At present, great attention is given to the training of personnel

strictly specialized in this area. This is carried out in a course for postgraduate training in "Manipulators and Robots" in two areas "Electronic Systems in Robot Engineering" and "Manipulators and Industrial Robots."

At present the first two groups of students are being trained. In addition to theoretical training, basic attention is being paid to practical work with the future specialists, and this is being carried out in the laboratories of the center; and not in last place are the scientific research activities.

Thus, our nation is establishing a new science and new production which will grow and raise the level of our technical development.

10272
CSO: 2202

CURRENT RESEARCH WORK AT UNIFIED CENTER FOR CHEMISTRY

Sofia VECHERNI NOVINI in Bulgarian 16 Feb 80 pp 1, 4

[Article by Academician Bogdan Kurtev, director of the Joint Center for Chemistry of the BAN [Bulgarian Academy of Sciences]: "Chemistry Aids Production"]

[Text] Throughout the Seventh Five-Year Plan joint activities carried out by contract with the Ministry of Chemical Industry; new photographic material for micro-electronics; by the end of 1980 enterprises in Sofia will achieve new developments; installation for the recovery of hydrogen sulfide at the Svilozha Economic Chemical Combine in Svishtov, with one-time savings of nearly 1.3 million leva.

The Joint Center for Chemistry of the BAN is a dynamic organization steadily seeking ways to upgrade the importance, quality and effectiveness of its activities, firmly following the party line in the field of science and creatively applying its principles. The Chemistry Department of Sofia University is functionally integrated with our joint center. We are convinced that it is only through joint efforts that we would be able to resolve the ever more complex problems which the party and the government set our chemistry in connection with the building of a developed socialist society. This five-year plan we are developing far more extensively our integration relations, this time on a contractual basis, however. We have a contract for cooperation with the Joint Center for Biology of the BAN, for a number of problems related to the development of chemistry and biology in the national economy could be resolved only through joint research. Throughout the Seventh Five-Year Plan we have operated on the basis of a contract with the Ministry of Chemical Industry, concluded as early as 1975. Recently a new contract was signed between the chairman of the BAN and the minister of chemical industry. It concretizes the problems to be resolved this year to help the chemical production facilities over the next 10 years. We have joint programs with the Central Institute for the Chemical Industry and others. We are drafting new joint programs with various departments of economic ministries.

As the coordinator for basic chemical research in the country, we have established close cooperation with the higher chemical technological institutes in Sofia and Burgas, and have set up a special chemistry coordination council for the VUZ in Plovdiv. At the same time, considering the limited nature of our scientific potential, we have developed very extensive, intensive and effective international scientific cooperation, above all with Soviet academic institutes, which involves individual colleagues, scientific groups and entire laboratories from VUZ and departmental institutes. This enables us to carry out scientific research on a higher level and on a considerably broader front.

The physical chemistry trend is the most intensively followed in our research. It is being developed by the Institute of Physical Chemistry--an institute enjoying very high international prestige--the Central Laboratory for Electrochemical Sources of Current, and the Central Laboratory for Photographic Processes. Last year as well a number of considerable new achievements were recorded in this direction, of few of which we shall mention.

The theory of embryonic formation, developed in the past by our physical chemistry, was applied originally and successfully in a new area--foam membranes. The results are of major importance in establishing the factors which determine the stability of various foam emulsions. We clarified the mechanism of action of some catalytic agents in the electrochemical reduction of oxygen. This will improve the quality of carbon-air gas-diffusion electrodes, which is one of the most important directions in the development of electrochemical sources of current (jointly with the GDR). Tests conducted on a parallel basis in the GDR and Belgium confirmed that the new photographic material being developed in our country with high discriminating capacity has photographic qualities which considerably exceed the best global accomplishments. This photographic material is used in microelectronics and is consistent with its latest requirements for condensing data registration. The extensive applied studies conducted this five-year plan in the field of galvanic films are being successfully developed, and by the end of 1980 will result in a number of applications which will resolve entirely and in several variants the two most important industrial galvanic processes: copper-nickel-chromium lining and zinc lining (jointly with the Institute of Organic Chemistry and the Central Laboratory for Polymers and, partially, with some scientific institutes and automobile plants in the USSR and some GDR institutes).

Another direction is the inorganic line developed by the Institute for General and Inorganic Chemistry, and by the sectorial laboratory for chemical reagents and preparations, which is a joint unit of the BAN and the Ministry of Chemical Industry. Successful work is being done in the field of inorganic synthesis, mainly the high-temperature synthesis of substances such as adsorbants and catalyzers, technical monocrystals, boron lining for increasing the durability of metal-cutting plates and instruments, and the synthesis and analysis of pure and superpure inorganic

reagents and other chemicals. A catalyzer for the complete burning of smoke gasses related to dump trucks is in the process of application. This is one of our original solutions which avoids the use of platinum; boron covers have been developed, rated "K," and winners of an international medal; inorganic chemical reagents have been developed and applied at the Krasnay Zvezda Factory in Vladaya Village.

The third direction is organic--general organic chemistry, bio-organic chemistry, kinetics and catalysis, and liquid and solid fuels developed by the Organic Chemistry Institute with a fetal chemistry center; and the synthesis and properties of polymers, developed at the Central Polymers Laboratory. A new emulsifying substance consisting entirely of local raw materials has been developed and its regular production has been undertaken. The Neftokhim SKhK in Burgas has applied a new technology for the processing of phenol resin, obtaining valuable semifinished products. A number of developments have reached an advanced stage. They will be applied before the end of 1980 by the Sofia Chemical-Pharmaceutical Plant, the Krasnaya Zvezda Chemical Reagents Factory, the Lakprom Enterprise, the new Dimitur Toshkov SKhK and others. A two-volume monograph on catalyzers has been published.

Our fourth direction is being developed by the Central Laboratory for Theoretical Foundations of Chemical Technology. This is very important in optimizing and intensifying chemical industry processes. However, the cadre potential and working premises of this unit remain quite inadequate. Despite this, however, its collective was able to achieve considerable successes. On the basis of a contract with the Svilozha SKhK in Svishtov, the available adsorption installation was restructured for the trapping of hydrogen sulfides. Here four inventions were applied, and the new system is superior to the best ones in the West. A one-time saving of nearly 1.3 million leva has been acknowledge. Also on the basis of original research a new rectification column was installed at the SKD in Dimitrovgrad, for separating multiple-component liquid mixes, insuring very high purity for the produced aniline (99.99 percent), successfully exported to the FRG and other countries. Such successes could easily be increased by applying them in other chemical industry branches.

The collective of the Joint Chemistry Center responded to the PCP Central Committee letter to the entire toiling people on the successful completion of the Seventh Five-Year Plan and taking a good start for the next new five-year plan. Today the implementation of these pledges is our most important task.

REPORT ON NEW PATENTS, TECHNOLOGIES, MATERIALS

New Patents and Inventions

Sofia ELEKTROPROMISHLENOST I PRIBOROSTROENE in Bulgarian No 11, 1979 p 430

[Text] Authorship certificate number 26,330. System for Visual Control of Analog Data. Engineer Ivan Kr. Kurtev, Doctor of Technical Sciences Yuliyar P. Marinov, Engineer Racho M. Ivanov and Engineer Asen O. Seizov, Sofia.

The system consists of a CRT [cathode ray tube] with deflection coils. It is distinguished by some characteristic features. The control electrodes of the CRT are related to the outlets of a block of current amplifiers.

The first information input of the block is connected to a digital-analog convertor for horizontal deflection, whose input is connected with a horizontal deflection register. The latter is linked through its control input to the control busbar of the microprocessor system. Through its starting inputs, it is respectively connected to the starting and recording circuits, and through its control outlet to a selection circuit which is linked with the control busbar of the microprocessor system. Through the starting input the control electrodes of the CRT are connected to the recording circuit, while its control outlet is connected to the address register, one of whose starting inputs is related to the recording circuit, while the other is connected to the address selection circuit. The address outlet of the address register is connected to the control circuit for horizontal deflection, linked through its control input to the control busbar of the microprocessor system, and with its control outlet to the address busbar of the microprocessor system. The other information input of the current amplifiers block is connected with the digital-analog convertor for vertical deflection. The input of this convertor is linked with the information outlet of the control system for vertical deflection, whose information input is related with the information busbar of the microprocessor system.

The system is to be used in medicine, electrical engineering, and so on. It offers the following advantages: Direct control of the CRT with deflection coils; possibility to select the necessary data, thus improving the man-machine link.

Authorship certificate number 26,346. Mono-oscillator. Candidate of Technical Sciences, Engineer Stefan A. Vulkov and Andrey Kr. Georgiev, Sofia.

The mono-oscillator consists of an RS trigger and input mono-oscillator based on the two-step OR/OR-NO. The non-inverting output of the circuit is connected through a capacitor with one of its inputs, and a resistor has been placed between the same input and the ground. The RS trigger consists of two two-step EITHER-NO circuits. The output of the first circuit is linked with one of the inputs of the second circuit, while the output of the second circuit is linked with one of the inputs of the first circuit. The mono-oscillator has several characteristic features. Thus, a resistor is placed between the output of the input mono-oscillator (inverting output of the logic circuit) and the input R (the free input of the first logic circuit) of the RS trigger. A capacitor is placed between the input R of the RS trigger and its output Q (the outlet of the second logic circuit). The input of the input mono-oscillator (the free input of the logic circuit) is linked with the S input (the free input of the second logic circuit) of the RS trigger.

The mono-oscillator may be used in pulse equipment. Compared with known similar mono-oscillators, it offers the following advantages: identical initial pulses in activating the mono-oscillator from a stable condition and its activation before the completion of the preceding cycle; faster action and a simplified circuit.

Authorship certificate number 26,246. Electronic Load Limiter for Electric Hoists. Engineer Stanislav St. Stanev, Engineer Stefan V. Golemanov and Engineer Nacho M. Nachev, Gabrovo.

The electronic limiter provides in one of the phases of the electric motor a low-ohm resistance and a systematically connected, after it, rising transformer, rectifier, a balancing link and relay element. Furthermore, it has a trigger with an emission connection and time-setting unit. It has the following specific features: The output of the balancing link is connected both to the input of the differentiating unit and to the input of the trigger, which has a time-delay element. The output of the differentiating unit is connected to an inverse booster-limiter whose output is connected with the module AND of the logic circuit. The output of the trigger is linked with the input of the second module AND of the same logic circuit. The time-setting unit is connected to the second inputs of modules AND of the logic circuit, while the output of both AND modules are connected to the logic element OR. The output of this element in terms of the output of the logic circuit is linked to the relay element.

The electronic limiter is to be used in electric hoists whose lifting mechanisms are powered by a cylindrical or cone-shaped synchronous electric motor. Compared with other similar electronic limiters it has several advantages, such as: the termination or bending of the cable or any changes in its characteristics in the course of time have been avoided.

These are shortcomings found in the limiter with strictionmagnetic gages and the electromechanical limiter; no special load bars are required as with the limiter using a tensiometric gage.

Authorship certificate number 26,344. Relaxation Generator with Pulse Series. Candidate of Technical Sciences, Engineer Khugo M. Oskar, Sofia.

The pulse series generator consists of two transistor-transistor logic integrated elements, whose outputs are the outputs of the generator. It is distinguished by several characteristic features. The feeding busbars of the two transistor-transistor logic elements operating within a breakdown system, are grounded, while their outputs are connected also to the first power source, respectively, through resistors. A time chain, consisting of a resistor, is connected between the input of these elements. One end of the resistor is connected to the input of the first transistor-transistor logic element. Consecutively connected, on a parallel basis the same resistor, are a capacitor and a power meter. The slider of the power meter is connected with the base of a key transistor. A resistor is connected to the emitter of this transistor and the input of the second transistor-transistor element and the gaging capacitor, whose other end is grounded, are connected to its collector. The other end of the resistor, the power meter, and the emitting resistor, are connected with a second power source.

The relaxation generator may be used in electronics and automation. Compared with known similar generators, it has several advantages: simplified design, easy control of the ratio of frequencies of pulse sequences within a broad range, and possibility to receive a step-like signal.

New Products, Materials

Sofia ELEKTROPROMISHLENOST I PRIBOROSTROENE in Bulgarian No 11, 1979 pp 431-432

[Text] 8R (TsAP-8R) Digital-Analog Convertor

The 8R (TsAP-8R) digital-analog convertor is used for converting an 8-bit twin-voltage code. The maximum value of the initial voltage with a single value of all bits of the input digital word is -5 volts.

The TsAP-8R includes an input coder, analog keys, a resistor matrix of the R-2-R type, an integrating amplifier and a source of reference voltage. The R-2-R resistor matrix is a thin-layer integrating circuit.

The TsAP-8R is made of standard electronic elements and is a reassembled module in a metal box, 109 by 86 by 12 millimeters. The outputs are placed in two rows on one side of the module, which permits easy assembly on a printed circuit board.

The TsAP-8R may be used in digital control systems, goods based on the microprocessor system, in systems for gathering, processing and controlling data of production processes, instrument manufacturing, and so on.

Technical Characteristics

Number of double bits--8

Input logic levels, standard TTL--s levels

Scope of initial voltage, 0 to -5 volts, and 0 to -10 volts

Absolute error in converting, $\pm 1/2$ LSB (± 9.7 millivolts; ± 19.5 millivolts)

Setting time: less than 20 microseconds

Existence of input buffer register

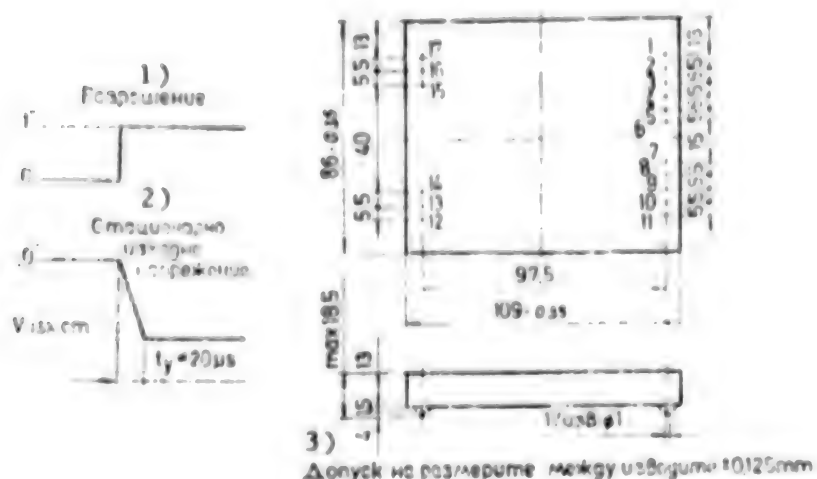
Power: ± 15 volts; 5 volts

Consumed power less than 3.5 volt-amperes

Working temperature range 0 to 10° centigrade

Storage temperature (-10 to 70° centigrade).

Figure 1 Time-Pulse Diagrams



1. Solution

2. Stationary initial voltage

3. Tolerance between outputs: ± 0.125 millimeters

Manufacturer: Microelectronics Institute, Sofia, Lenin Blvd--7th km
Telephone: 71-21-41; Telex: 22064

Exporter: Izotimpeks, Sofia, 51 Chaparev St
Telephone: 63-71; Telex: 022731

Thermoregulator for Radiators ESPA 01 AO 01 and ESPA 01 AO 02

Used for controlling temperature in housing and working premises within certain limits through changes in the heat release of water-heating radiators through changes in the flow of hot water running through the thermoregulator.

As the temperature of the air changes the piston of the sensor of the radiator is activated as a result of changes in the pressure of saturated vapor. This acts on the axis of the valve, which opens partially or fully the inlet. The control is proportional. Turning the tuning section, the piston shifts in terms of the axis of the valve, thus regulating the temperature.

Technical Data:

Type of valve: angular, in ESPA 01 AO 01
straight in ESPA 01 AO 02

Range of tuning: 15 to 25° centigrade

Full range of the sensitive element: 4-7 millimeters

Maximum water temperature: 100° centigrade

Maximum pressure: 0.6 MRa

Testing pressure: 1 MRa

Dimensions: B-- ϕ 68, H--128.5, Pipe--1/2"

Produced by the Instrument Manufacturing Combine in Petrich

Indicator for Storage Battery Discharge IRAB 12, 24, 40, 80

The indicator indicates the level of discharge lead batteries through light signals. It is an electronic instrument which turns of a signal bulb set on the outside, containing a heated wire which changes conditions according to the voltage in the storage battery. The results are the following:

Continuing light signal--drop of voltage below U_2 .

Threshold voltages U_1 and U_2 may be set within the following limits, as requested by the consumer:

U_1 --from $0.80 U_V$ to $0.90 U_V$

U_2 --from $0.6 U_V$ to $0.75 U_V$

Nominal voltage--12, 24, 40 and 80 volts

Reduced error of threshold voltages at outside temperature of $20^\circ \pm 5^\circ$ centigrade and relative humidity not to exceed 90 percent, not exceeding ± 4 percent.

Insulation resistance at a temperature of $12 \pm 5^\circ$ centigrade and relative humidity ranging from 20 to 80 percent--higher than 3 million [3 M] ohms.

Dielectric strength of the insulation--1,500 volts, 50 Hz, for 1 minute

Produced by the Instrument Manufacturing Combine in Petrich.

SESB Cable

In 1978 the V Kolarov Cables Plant in Burbas mastered the production of the SESB cable. It is a symmetrical main cable with styroflex cord insulation, used for main cable and connecting lines for high frequency packing, with ranges of up to 250 kHz and 552 kHz.

Four power conductor cores insulated with styroflex cord and wrappers of different colors are spun in star-shaped foursomes with a different pitch. The foursomes are twisted in a cable bunch and paper insulation is added. A lead shielding covers the cable bunch. It consists of two steel ribbons and an external cable yarn cover.

The cables are based on the stipulations of state standard 15,125--1976.

The electric parameters at a temperature of 20° centigrade must be the following:

Active resistance of the cores per 1,000 meters, 15.85 ohms

Ohmic asymmetry of the cores in the working pair, 0.19 ohms

Insulation resistance of each core compared with the others, linked with the metal shielding--12K megaohms [12,000 M] per 1,000 meters

Operational capacitance for 1,000 meter length at a frequency of 800 Hz, 24.5 +/- 0.8 nanofarads [nF]

Crosstalk attenuation of the near end between each couple 825 meters long at a frequency of up to 252 kHz, for 100%--59 dB; for 90%--62.6 dB

Shielding of the far end between all couples, 825 meters long, at a frequency not exceeding 252 kHz, for 100%--67.9 dB; for 90%--73.9 dB

Capacitance connections $K_{2,3}$ and capacitance asymmetry $e_{1,2}$ for a length of 825 meters: for 100%--580 picofarads [pF]; for 90%--230 picofarads

Packet Cam Switch Breaker PGP 10-03-20 and PGP 25-03-20

These instruments are used for turning on, stopping or reversing asynchronous electric motors and for controlling all types of AC chains--power or command with fixed current, corresponding to the nominal current of the given instrument.

The new packet cam switches and circuit breakers have nominal currents of, respectively, 10 and 25 amperes, with nominal tension of 380 volts. They are used for ASz application, as per Bulgarian state standard 5,329-76.

The circuit breakers will be marketed with IP-00 and IP-54 (socket outlet only) shielding and IP-54 protective lining as per Bulgarian state standard 3,440-1965.

Nominal current, 10 and 25 amps

Nominal tension, 380 volts, 50 Hz

Displacement voltage, 660 volts

Switching capacity: ASz

Mechanical resistance: 3×10^6 breaking cycles

Electrical resistance: 5×10^5

Operational regime: continuous and repeated short time with PV not exceeding 60% with circuit breaking frequency of up to 600 per hour

Operational frequency of switching: up to 300 switching cycles per hour

The circuit breakers meet the requirements of Bulgarian state standard 4004-1973, 5329-1976, the MEK 408 1972 publication, and Bulgarian state standard 3440-1965.

They may be produced for ships with an unlimited sailing range and cruising in tropical waters.

Producer: Instrument Making Combine in Petrich.

Activities in Institutes, Enterprises

Sofia ELEKTROPROMISHLENOST I PRIBOROSTROENE in Bulgarian No 11, 1979
pp 433-44

[Text] Production of Class B + C Amplifier. By Docent Candidate of Technical Sciences Engineer Vasil B. Vasilev.

For the past two years a group of young specialists at the Chair of Radio Engineering of the Lenin VMEI [Higher Machine-Electrical Engineering Institute] in Sofia has been intensively working on the practical develop of a powerful, highly effective and high-quality class B + C amplifier for use in cable radio transmission.

On the initiative of the Elektroakustika Plant in Mikhaylov, and the IRE [Institute of Rationalizations and Experimentation] in Sofia, a contract was concluded at the beginning of this year among the plant, the VMEI and the IRE on the mass production of the highly effective B + C amplifier. The contract became a major incentive for scientific research and experimental work at the VMEI.

At the end of August, in honor of the 35th anniversary of the founding of the Lenin VMEI, the first amplifier prototype was completed and tested. The resulting power and quality indicators exceeded the stipulations of the sectorial norm ON-09 69368 for superior-grade power amplifiers.

Power and quality indicators and effectiveness:

Net line power, 150/160 watts [W]

Harmonics coefficient in the 0.09-0.29% range

Frequency band from 40 Hz to 16 kHz +/- 1.5 dB

Effectiveness: For signals with a level of about 50% of maximum (at which level the amplifiers can be operated over the longest period of time), the industrial effectiveness of the amplifier is about 150% (i.e., a factor of 2.5) higher than of the most extensively used amplifier of grade B power. For higher amplitude signals the effectiveness values are considerably higher than those of grade B.

Such high effectiveness of the amplifier has made it possible to drastically reduce the volume, weight, cost and operational expenditures of domestically produced amplification systems TUV-600T and TUV-1000T, in which the 100 watt grade B structural unit will be replaced, with the same volume and weight by a 150 watt B + C class unit. The expected annual savings will exceed 800,000 in foreign currency.

The amplifier has been recognized as an invention in the Bulgarian People's Republic.

A prototype of the amplifier was an item of the exhibit "The Lenin VMEI--as Old as the Socialist Revolution."

Chengeliyev Resistors Plant in Aytos. By Engineer Nikola I. Nikolov and V. Popova.

The plant was founded in the 1960, specializing in the production of low-voltage items. It became operational in 1961. Its main production items were brushes for electric power machinery for automotive transportation, lightning rods and silver points. In 1962 it mastered the production of carbon and wire resistors, which became the plant's main production items. Five years later--in 1967--it mastered the production of potentiometers in a new production unit. In 1969 the industrial carbons shop was moved to Kableshevo. In 1970 it undertook the production of permanent metal plate resistors based on Soviet technical documentation and undertook its automation.

Currently the Chengeliyev Plant in Aytos is the only plant in the country producing direct and alternating current resistors (resistors and potentiometers).

The following items are produced by the plant:

DC laminated metal resistors of the RPM-2 and RPM-3 type with a power of 0.125 watts [W], 0.25 watts, 0.5, 1.0 and 2 watts:

DC carbon laminated resistors of the RPV-4 type with a power of 0.125, 0.25, 0.5, 1 and 2 watts;

DC wire resistors--bare, lacquered, enameled and controlled;

AC composition resistors (potentiometers) of the PK type, with a power of 0.1, 0.25 and 0.5 watts;

Tuning composition resistors of the DK type with a power of 0.05, 0.25 and 0.5 watts.

The plant's output meets essentially the requirements of the radio electronic, communications and computer equipment in the country. Resistors

are exported to the USSR, Hungary, France and Italy, and, this year, to India and Sweden.

Immediately following the creation of the plant a development and application base was set up with two bureaus in charge of developing new items--a DC resistors bureau and an AC resistors bureau.

The basic task of the BRV [Development and Application Base] is to develop and apply new improved items and technologies in resistor manufacturing. It determines the most effective nomenclature of the plant. Thus, for example, this five-year plan, the average rate of new and improved goods has been 10%; a 14% figure was achieved in 1978. In 1978 one new technology and six new items were introduced, one of which was over and above the plan. More topics and higher quality indicators were introduced in 1979.

Currently work is underway on mastering AC pretuning resistors of the metal ceramic type. The purpose of the use of this license is to resolve the problem of ceramic quality.

The main line of development of the Chengelievi Resistors Plant is integration with the Soviet Union in the field of AC resistors. This will lower the great variety of produced resistors. Integration will lead to the development of several mass-produced items. This will raise the level of production mechanization and automation and the plant's capacity will nearly double.

In 1977 pretuning laminated 0.25 watt SP 3-1a and SP 3-1b type resistors were manufactured on the basis of integration with the Soviet Union. Such resistors have a better design and are more reliable. They have replaced the old designs of pretuning 0.25 watt resistors. They are produced for the domestic market as well. The plant will master the production of AC wire resistors of the SP 5-2 type and of precision metal laminated resistors for the domestic and export markets. Automated assembly lines, imported from the USSR, were installed for the production of DC metal and carbon laminated resistors, and Soviet technology is used.

One of the tasks related to integration processes and to upgrading the share of AC resistors is the development of a technology for the manufacturing of resistor mixes based on Soviet materials, and their application on Bulgarian getinax for the production of AC pretuning resistors. The solution of this problem will stop imports from the capitalist countries.

The base has a unit for technical and patent information and a prototype workshop.

Both the base and the plant are closely linked with related enterprises in the USSR, such as, for example, the Resistors Plant in Odessa and the First Moscow Radio Parts Plant.

The plant has a testing laboratory which includes reliability testing as well. All basic items produced by the plant are certified as first category, bearing the quality mark "1."

Following are some basic economic results characterizing the enterprise's activities:

The 1978 plan for overall industrial output called for 6.1 million leva; the factual output was worth 6.3 million leva;

For the same year the planned net output was 3,209,000 leva; the factual output was worth 3,466,000 leva;

The volume of "1" rated output was planned at 3,341,000 leva; the factual amount was 4,510,000 leva.

Currently the main direction followed by the plant is to apply leading experience in all shops and extensively apply new forms of labor organization in order to raise social labor productivity. Thus, for example, all units are following the Lvov system and the experience of the Lithuanian SSR. The shop for DC metal laminated resistors works on the basis of the experience of the Nayden Kirov Plant in Ruse; the shop for potentiometers follows the initiative of the Chemical Plant in Vidin. In addition to these initiatives, the plant is proud of its own. The main objective of the labor collective is "from high work quality of the individual to high effectiveness of the work of the collective."

5003

CsO: 2202

POSSIBILITY OF COMPUTERS BASED ON NON-NEUMANN PRINCIPLES

Budapest INFORMACIO ELEKTRONIKA in Hungarian No 1, 1980 pp 3-9

[Article by Gabor David, a scientific team leader at the SZTAKI, Computer Technology and Automation Research Institute of the Hungarian Academy of Sciences: "Possibility of Computers Based on Non-Neumann Principles"]

[Text] One possible version of computers based on non-Neumann principles is a computer based on higher order logic. The author introduces the architecture for such a computer which is illustrated by a scheme analogous to a "classical" Neumann computer. He provides logical languages corresponding to the language for data and programs for the description of the facts (information) and the problem solving strategy and then introduces a possible realization of this computer.

Introduction

The conception for a Neumann computer--in the first study by Neumann, Burks and Goldstine and in the computers developed since then--is based on a digital (binary) technique. The essence of this technique is that both data and programs are stored in a binary representation; the basic logic and arithmetic operations are made up of operations in Boolean algebra. A reformulation clarifies this fact in its more profound interdependencies:

First Thesis

"The basic operations of a Neumann computer are operations of Boolean algebra and Boolean algebra is a model of zero system logic--adding that in essence it is an unambiguous model with isomorphic power."

A zero system logic language is suitable for the formulation of simple statements; it contains predicate symbols and sentences obtained by linking them in an appropriate way with logic operations. For example:

$$((\sim P \vee Q) \wedge R) \rightarrow Z$$

where $\neg$ is "not," $\vee$ is "or," $\wedge$ is "and," $\rightarrow$ is "implies" and P, Q, R and Z are predicates which signify the existence or non-existence of some property. The Neumann computer performs an evaluation of such formulas when the possible values of the predicate symbols, logical truth or falseness (that is, 1 or 0) take the place of the predicate symbols. There are two very important consequences of this conception:

--the computer can work only with previously defined concrete data and instructions, and

--the execution of each instruction does not depend on the preceding ones; one can always say what the result will be in each case.

This definiteness, which is found in both assertions, is quite different from two very important factors in human thought, namely the formulation of concepts, including the use of concepts, and learning. (We do not want to discuss here the unity and difference of these.)

In man-machine relations the Neumann computer always involves this definiteness and so it forces on man a form of communication which is more or less alien to him. This level is raised ever higher by the languages of ever more complex software systems but behind the conversation between man and machine always stands the fact that the computer is reducing this conversation to the level of zero system logic, which is to say that it can express only what can be reduced to the initial level.

Software system planning and the planning of systems in general was the first level of applications where the above, frequently well hidden restrictions appeared as an unmistakable obstacle. Someone designing a system is working with concepts and objects which cannot be precisely formulated and yet he has the requirement, in every phase of development, of making the system under development surveyable, proving whether it is right (by simulation or proving or with tests) and of helping the computer to go on. The thinking or reasoning of the engineer is always hidden to the computer because it cannot be formulated economically in the "reducible" form as defined above.

It is obvious that the nature of the problems belonging to the theme of "artificial intelligence" is similar to the case illustrated with the problem of the system designer. The solution to these problems with the best prospects is suggested by the formulation given above in the First Thesis, which we now reformulate:

Second Thesis

"The basic operations of a non-Neumann computer are the operations of some 'higher order' algebra and this algebra is a model of a higher order logic."

We formulate this assertion thus so that the analogy with the First Thesis will be complete. At the same time it provides two directions for research:

--we must be able to handle those logical languages which can express in an adequate way the problems in the various branches of science and in the light of which we can execute the hints of the Second Thesis, and/or

--we must search for material phenomena which can store more complex algebraic structures and perform operations on them.

For the time being we consider the first of these two the more important--to find a logical language which stands close to the language of the experts and on the basis of this to provide an efficient computer for the user field.

In what follows we provide a general scheme of a computer fitting the Second Thesis and then we will summarize the results achieved in a concrete direction.

We cannot at all say that this research is completed or has even led to spectacular results. Many research tasks remain in both the general scheme and in the concrete versions but on the basis of research thus far we are convinced of the correctness of the Second Thesis and of the utility of a computer designed for this purpose.

From this point forward we want to abandon the term "non-Neumann" and will refer to it as a "logical" computer.

The General Scheme

In every thought process concepts appear together with judgments; the link, difference or co-existence of some concept with some other concept or concepts combine to give the essence of the things covered by the concept. These judgments or assertions will play the role of facts. We draw new facts from these facts, according to the given rules of thought.

What is it that corresponds to facts and conclusions in the case of a logical computer? As always, we must now return to an analysis of a Neumann computer.

In general concepts do not exist in a Neumann computer; only the user can give such an interpretation to its several objects. Data play the role of facts; the conclusions drawn from them, that is the generation of new facts or data, means the execution of program instructions. The program as a whole is the solution of the problem in which the user specifies how or through what data series we get the answer to the given problem. In accordance with this we give the scheme of a Neumann computer in Diagram 1 (deliberately separating programs and data in memory and deliberately leaving out the input and output components).

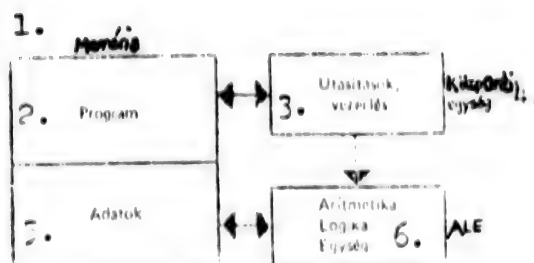


Diagram 1.

Key: (1) Memory (4) Central unit
 (2) Program (5) Data
 (3) Instructions, control (6) Arithmetic, Logic Unit (ALE)

The several instructions of the program always go into the central unit and these control the data transformations, that is the generation of new data, to be performed by the Arithmetic/Logic Unit or ALE.

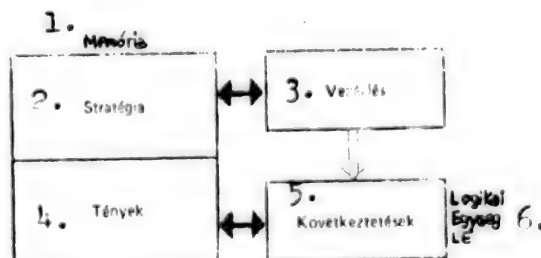
In the case of the logical computer--using the analogy only for the sake of better understanding--the scheme is essentially the same as in the case of the Neumann computer but we must reformulate the meaning of the several components.

Third Thesis

Diagram 2 is the scheme of the logical computer where the logic unit or LE draws the conclusions concerning new facts, the control selects from the facts those to be used and receives the new, derived facts while the strategy, in the course of problem solution, describes the heuristic, thought schemes to be applied (that is, what type of conclusions the user expects from what type of facts and, if he does not get them, what is to be done then). The strategy is closely connected with the control because the control is nothing more than the implementation of certain elements of the strategy.

Diagram 2.

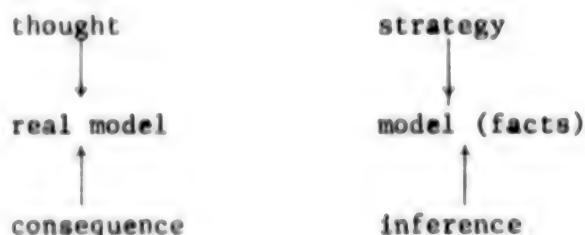
Key: (1) Memory
 (2) Strategy
 (3) Control
 (4) Facts
 (5) Inferences
 (6) Logic Unit (LE)



Every single element of the scheme formulated in the Third Thesis requires profound analysis and research. Thus the language of the "facts" is the logical language to be used for the given task and the usual (or unique) methods of inference for the task must be defined by the strategy and the control so that the logic unit can carry out the transformations taking place in the real model. This immediately brings up the necessity of the identity of the inference being used in the logical language and the "concept of inference" being realized in the real model (adequateness).

Even at this level of generality (which admittedly does not say much) it must be briefly explained what it means to use a logical computer. We can express its purpose with the expression "modelling and problem solving aided by a logical computer."

The facts describe some part of reality. Using mathematically imprecise words, the real model is the part of reality to be examined and the "internal reality" expressed by the facts is a model of the real model. The language of model creation is the language of facts. This is called upon to reflect the real model (which obviously means at the same time a high degree of reduction or simplification).



In the real model there is the concept of consequence (describing cause-effect interdependencies and transformations) and logical inference is valid in the model described by the facts. Thus thought uses both consequence and inference to recognize processes taking place in the real model; that is, it recognizes consequences from reality in its thoughts while at the same time using the logical operations of inference in a model described in exact language. We should emphasize two aspects of this:

- it makes hypotheses, and
- it checks them.

The making of hypotheses belongs to the sphere of human thought although several attempts have been made to mechanize them^{1,2} but the checking of hypotheses can be done in the model by the logical computer, namely it can prove the correctness of assertions. The strategy controls the course of the proof.

A Concrete Direction

We noted several times in the foregoing that supporting the preparation and development of computer systems was one area where the use of a logical computer and inference according to the above scheme promised success. We do not want to analyse the problems of this field here; it is sufficient, on the one hand, to refer to the "software crisis" and, on the other hand, to note that technological development has made possible the development of a great variety of special systems and special computers, so we are approaching the age of a "hardware crisis."

In the beginning this direction came up as the automation of microprogramming and led to the Structure Logic or SL language^{3,4,5} which serves to describe the operation of computers. The language is suitable to describe and specify the operation of programs since the basic idea of it is a description of the data structures in the computer (in such a way that the several elements of the structures are provided with types) and a description of the operations performed on these structures. If we disregard strategy one could build up on this basis an automatic proposition prover (which stores only the facts, having a logic unity and a "wired" or fixed control) which constructively proves the existence of the desired program--namely, provides that program which carries out the desired operation. The logic base for the Structure Logic is many-sorted logic, supplemented by a special type, the selector (and its models are heterogenous algebras, also supplemented by the selector). The SL language can be used not only to describe concrete, hardware level architecture but also abstract processors of high level languages (see the articles by Burstall and Coguen⁶, David⁷, David and Sarkozy⁸ and others).

It has been shown that this language can be used to describe the structure of parallel operating abstract or real computers.⁹ We will briefly describe the SL language here. We will put the emphasis on a description of the strategy.

SL--The Language of Facts

With the language we can describe the types of objects in the real model, the structures which can be built out of them and the transformations interpreted by them. (A description of the language can be found in reference 4.)

In order to define facts in SL one must define those complex structures which represent the problem. The user can define basic types the interpretation of which does not have to be given to the logical computer (that is, they do not have to be defined at a lower level). Complex types can be defined from basic types or from already complex types:

structure type name $\langle s_1 : t_1, s_2 : t_2, \dots, s_n : t_n \rangle ;$

where $s_1 - k$ are selector symbols and $t_1 - k$ are already defined types. The type thus defined represents those things which have n components, each in

order connected with $s_1, s_2, \dots, s_n$ selectors and each in order having $t_1, t_2, \dots, t_n$ types. Homogeneous types, those which contain n of the same t type part types, can be built up in the manner

structure type name $\langle [1 : n] : t \rangle$.

We can declare symbols which represent objects corresponding to the given types:

reference (type name) symbol list.

The basic types must include those transformations which take place through the type structures in question and at the time of declaring this basic type must be described as a part type as the structure of the appropriate (complex) type. For example, if the instruction and the bit (8) is the basic type then:

structure ALU $\langle 1 : \text{bit (8)}, 2 : \text{Register}, 3 : \text{instruction} \rangle$;

structure Register $\langle [1 : 8] : \text{bit (8)} \rangle$;
instruction move (1), store (1), add (1):

in the declaration the ALU "type things" include those operations which transform these (namely, in the third part type).

The reference (ALU) ARITM;

declares the ARITM symbol, as it represents ALU type things.

We call those symbols variables which are not declared. The type of the variable is that type in whose place it appears. Symbols declared by type and handled as constants can be declared with the basic word constant. The function symbols are special symbols where both the function and its parameters have types.

What has been said thus far provides a tool for the description of concepts and their structural links. These were definitions preceding the description of the facts and their effect will be reflected in the logical computer as a whole.

We describe the facts in the form of assertions. Assertions are sets of sentences made up of structure expressions with the logical symbols $\sim, \wedge, \vee$ and $\rightarrow$, usually of the form

$$S_1 \rightarrow S_2$$

where S_1 and S_2 are structure expressions. The general form of the structure expression is

$$S \langle s_{i1} : 0_1, s_{i2}, \dots, s_{ik} : 0_k \rangle$$

where S is a declared symbol of some type w , $s_{1j} \dots s_k$ is some selector figuring in the w declaration (all different) and the $O_{1j} \dots O_k$ are the symbols of the appropriate t_{1j} types. The interpretation of a structure expression is that the s_{1j} th component of the thing represented by the symbol S contains O_{1j} where j equals 1, 2, ..., k while the others are discretionary (that is, those parts selected by the s_{1j} 's for which $1 \neq 1$, and $1 \in 1, n$). We can call the part of S selected by the s_{1j} the $S[s_{1j}]$, or in the case of multiple access, in the form $S[s_{1j}][s_{2j}] \dots$ etc., it means the part selected by $S[s_{1j}][s_{2j}]$.

Continuing the above declaration, an example of assertions is:

ARITM < 3 : move (i) > $\rightarrow$ ARITM < 1 : ARITM [2] [1] >

The significance of this is that the left side says that the third component is a move (i), which is of the instruction type, and the other components are discretionary. The right side says that 1'th component of the second component goes into the first component, and the rest are discretionary.

The assertion as a whole says that the move (i) transformation carries the left side to the right side. The symbol $\rightarrow$ corresponds to the consequence concept of the real model and it is also the logical sign for inference (adequateness).

The facts can be described with such assertions. We would like to emphasize that the description in this form gives only the relationship of the appropriate symbols to one another; what these signify, that is what concepts they involve, depends on the user. There is no obstacle if, in the above example, someone interprets the bit (8) symbol as a memory of 1 Mbytes--if the assertions are valid then it can be justly so interpreted. Similarly, the operations performed on the structures could be symbolically described subroutines, programs or program systems--the effect of these can be formulated in the language of facts.

The Logical Unit

The logical unit derives new facts from facts in mechanical way. For the most part the basis for a mechanical proof of propositions in the principle of resolution, and this is true in SL too. The basis is this, that if two assertions contain the same structure expression with a negation in one and no negation in the other then we say:

$$A_1 : \neg S_1 \vee S_2$$

$$A_2 : S_1 \vee S_3$$

and the logical consequence of these two assertions is the so-called resolvent, that is:

$$A_3 : S_2 \vee S_3.$$

Naturally, the resolvent cannot always be derived in a direct form. This is why we need substitution.

In any assertion we can substitute one symbol for another if both have the same type and, in addition:

- a variable can be replaced by a constant, another variable or a function,
- a function can be replaced by a function, if they have the same names and their parameters can be substituted,
- a function can be replaced by a constant if the value of the function is the constant.

Substitution must apply to the entire assertion; that is, either we replace the symbol every time it occurs or we replace none. If δ is a substitution then we call the expression derived by substitution from the structure expression S the expression $S \delta$. The $S \delta$ is the logical consequence of S .

Using the principle of resolution, if there are two assertions

$$A_1 : \sim S_1 \vee S_2$$

$$A_2 : S_3 \vee S_4$$

such that there exist substitutions δ_1 and δ_2 where $S_1 \delta_1$ and $S_3 \delta_2$ are the same then the assertion (the resolvent)

$$A_3 : S_2 \delta_1 \vee S_4 \delta_2$$

is the logical consequence of A_1 and A_2 .

It is the task of the logical unit to find the suitable substitution and generate a new assertion. Since to perform substitution one must know the types, symbols and functions figuring in the model this unit of the logical computer must contain the declarations written in SL.

But we should not limit the logical unit to the generation of resolvents. The SL can be axiomatized as logic and the several axioms are all different operations for the logical unit. In what follows we will designate all these operations by the letter σ .

Control and the Description of Strategy

Insofar as we have set up our model, that is described it in the language of facts, then the proof of any assertion S can take place simply by starting from the opposition of the assertion to see if the system can show a

contradiction (indirect proof). Together with the facts already described this is the inference of new assertions and if in a sufficiently long chain of inferences there occurs an assertion for which there is no model, that is if a contradiction is generated, then we have proven the original assertion S . The symbol for that assertion for which there is no model, the empty assertion, is Π (nil). We designate the resolvent of assertion A and assertion B with the symbol $A \circ B$ (there being no assurance that it exists!).

If we designate with an I the set of assertions describing the facts and its elements are the assertions $I_1, i \in 1, n$ then the first step is to generate the resolvents with these, that is the $I_1 \circ \sim S$ assertions then for each one of these we generate the resolvents $I_1 \circ^1 (I_1 \circ \sim S), j \in 1, n$ and so forth. If at any level we reach an Π assertion then we have proven the assertion S and the proof is simply the series of facts leading to it (see Diagram 3).



Diagram 3.

This method leads to a combinatory explosion, so it is not useful to use it. The role of strategy is simply this, to define within this "search tree" a restricted, more useful search guided by human intuition. Naturally it may be that for any given problem we will not reach a solution in this way while some other strategy might have led to a result. The professional literature provides different strategies the completeness of which (that is, that they will always find a solution if one exists) is frequently very difficult to prove.

It was our purpose to analyze the general structure of strategies and to provide the chief characteristics of a language with which the user could formulate his solution strategy.¹⁰

The strategy describes the generation of a set of new assertions from sets of assertions. If S_1 and S_2 are two sets of assertions then $S_1 \circ S_2$ designates the new set which consists of the elements

$$\{S_1^1 \circ S_2^0 : S_1^1 \in S_1, S_2^j \in S_2\}.$$

Any strategy:

--does not necessarily select out every S_1^1 and S_2^j element,

--does select the logical operation to be applied, which we designate here with the letter o,

--does not necessarily accept every element of the form $S_1^1 \circ S_2^j$.

If we designate the initial set of facts as AX and the assertion to be proved as S, then:

$$\text{step } (0) = \{\sim S\}$$

$$\text{step } (n) = \{AX \circ \text{step } (n-1)\}$$

together describe that "strategy" which we showed in diagram 3 with the generation of resolvents. Here we perform resolvent generation on every element and we accept every element.

We thus conceive of the strategy as a recursive definition of sets and so we can use the language of mathematics. If there is to be a "programming language" then we must make a few additions.

The strategy is successful if we derive the Δ assertion, and then the entire system stops. The failure of the strategy cannot be formulated in so simple a manner. For example, we might call the above strategy a failure if it is not successful in 30 steps. Thus it may be a failure depending on when the programmer calls it a failure. The basis for this is

the failure condition,

where the condition is some assertion pertaining to the assertions (see below). So a strategy is a failure if the set $S_1 \circ S_2$ is empty, that is it cannot generate new resolvents either because none exist or because we do not accept them.

One can define in the strategy to the generation of what new set we should turn in the event of failure. The general form of this is:

the "on failure" set name, or set definition.

If a set generation which has been initiated does not succeed then we can write a new "on failure." The set name is a string which can have parameters and it can be made equal to a set definition, in the form:

set name = set definition.

The set definition is a set definition in the mathematical sense, that is:

$$\{ \text{typical element} \mid \text{constraints} \}$$

for example:

$$\{ \text{AoB} \mid \forall A \in \text{AX}, \forall B \in \text{step (n-1)} \}.$$

The parameters always include how many set generations we should conduct (that is, at what level we are in the search tree). For the sake of simplicity it need not be declared separately and it is always the first parameters. Control always increases it by one if the set generation was successful at the level in question.

The "strategy" for selecting from among the elements of the sets is either described among the constraints or takes place with a base word "new." We can define for any set how we will select elements from it. If Z is a set of assertions then

$$\text{new Z} = \text{constraints}$$

describes the relationship to the not yet selected elements of Z. This again brings up the necessity of being able to formulate assertions about certain properties of assertions written in SL. So we need a logical language in which constants, variables, functions and predicates are interpreted in assertions in SL.

An assertion constant is any assertion written in SL. An assertion variable is any symbol which we can refer to as an element of a set. We get a special assertion variable in the form NewZ, the momentary value of which is the assertion selected from the assertion set Z. Assertion functions are functions interpreted in assertions. These may be:

V (A) the number of variables in assertion A

C(A) the number of constants in assertion A

N (A) the deepest level of embedded functions in assertion A

L (A) the number of predicates in assertion A, etc.

Control knows these for every assertion. The functions have values and these can be compared. Predicate symbols are discretionary symbols with which we can describe properties pertaining to assertions. For example:

IMPLIES (A, B) true, if $A \rightarrow B$

EQU (A, B) true, if $A \leftrightarrow B$

EQUST (A, B) true, if $A = B$ (string)

ELEM (A, B) true, if there is a substitution for A which is equal to part of B.

Examples of their application are:

failure EQU (A, B) $\forall (n \geq 30)$

newZ : = (V (newZ) $\geq$ V (s) $\forall s \in Z$)

H = { A | A $\in$ K, $\forall A \exists B (B \in AX)$ IMPLIES (B, A) } .

We use set operations in the usual mathematical sense and with designation. It may be necessary for the strategy to generate sets step by step, sets which are not intended as notation for the course of the resolution. These are contained in the stem to be executed as the n'th step of the strategy. The stem is executed only if the strategy is not unsuccessful in that step.

The general form for describing the strategy is:

```
strategy-name:  initiation;
                n set definition
                strategy stem in the n'th step
                on failure...,
                .
                .
                .
                ;
                failure condition;
```

A; the several sections are separated and the comma is the separator therein. The first strategy can be written as follows in this form:

TRANSVERSELY:

```
step (0) = {  $\sim S$  } ;
step (n) = { AX o step (n - 1) } ;
failure (n  $\geq$  31);
```

Here the stem and the "on failure" action are empty.

That strategy which performs the search in depth, that is reaches a certain depth by repeated resolvent generation with the original facts and fails, then returns and selects a new element from AX, from the set of original facts:

LONGITUDINALLY:

```
new AX: = { N (new AX)  $\geq$  N (s) |  $\forall s \in AX$  } ,
level (1, C,  $\sim S$ ) = { Co  $\sim S$  | C = new AX } ,
```

```

• AXIOM (1) = AX;
  (end of initiation)
  level (n, A, B) =
  { AoB | A = new AXIOM (n)  $\wedge$ 
     $\wedge$  B  $\in$  level (n - 1), C, D) } ,
  AXIOM (n + 1) = AX,
  level (n + 1), newAXIOM (n + 1),
  AoB } ;
  (end of stem, we rise to a new level, where we run another element
  of set AX to the end and "new" is defined thereby)

  on failure level (n, newAXIOM(n), B), (that is, we select a new one
  and try to go on with it)

  on failure

  level (n - 1, newAXIOM (n - 1), D),

  (and if it does not succeed then we go back one level)

  failure (n  $\geq$  50)  $\vee$  (n = 0);

```

This strategy is well known, but we have modified it here in that we selected a "new" at every level according to the N function (the number of variables). This was necessary in order to get a unique example of the original A set at every new level so that we could note what elements we were testing at the given level.

The "on failure" instruction is needed twice; first to try another assertion at the same level--to see if we could proceed to another level or had to return. If we had to return this was dictated by the second "on failure" instruction.

We certainly cannot say that we have listed every essential element of the strategies. One deficiency might be that we have dealt here only with the "pessimist" strategies (that is, indirect proof) but the other direction could be described. Some concepts we have only illustrated without precisely defining them, but we did this deliberately.

The Road Leading to Realization

Diagram 4 shows a schematic of the information and control routes between information described in the language of facts, the strategic language describing the thought schema and the Logic Unit. Starting from step zero of the strategy the Control Unit executes the appropriate stem and then goes to the next step. This is indicated by a program counter. The execution of each stem is as follows:

--formation of certain sets from the facts,

--selecting assertions from the sets thus formed (execution of the "new"),
 --evaluating the assertion functions and the assertion predicates,
 --selecting the step to be performed by the Logic Unit,
 --analysis of the new assertion received from the Logic Unit and, if it is accepted, listing it among the facts.

One can store together with every fact the values of the assertion functions defined by the fact (the number of variables, the number of selector variables, etc.) and what sets it is listed in. With the aid of this we can search among the facts in an associative way according to set and property or we can cite them by their address (the assertion constants).

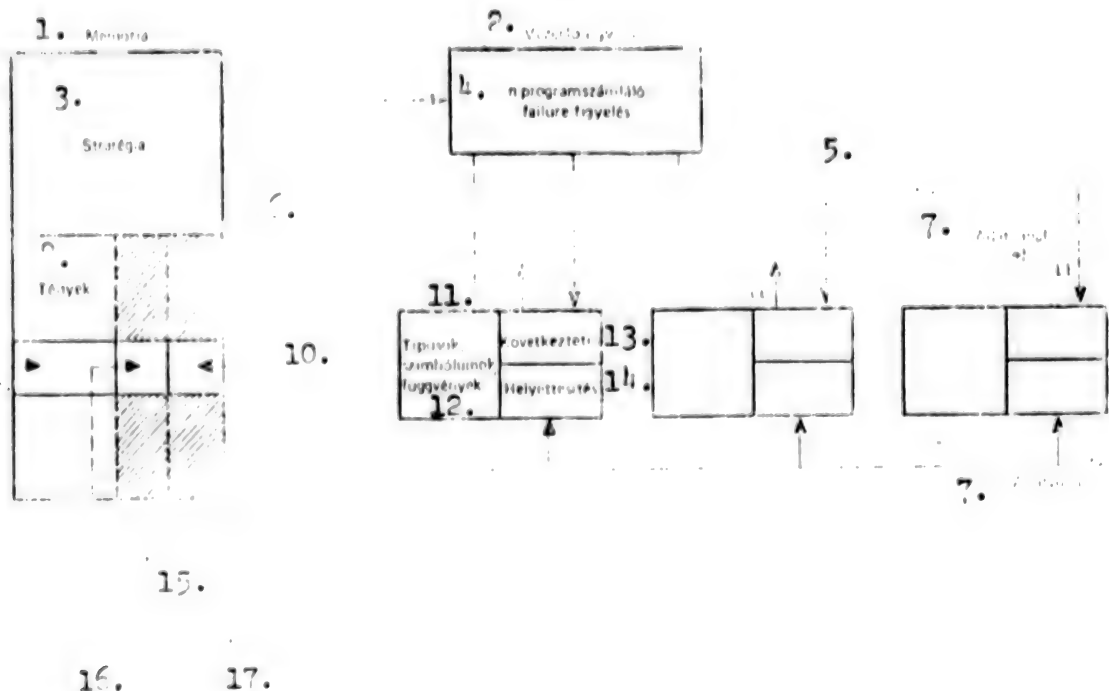


Diagram 4.

- Key:
- | | |
|--------------------------------------|------------------------------------|
| (1) Memory | (10) "New," set generation |
| (2) Control unit | (11) Logic unit (LE) |
| (3) Strategy | (12) Types, symbols, functions |
| (4) n program counter, failure check | (13) Inference |
| (5) Selection of inferences | (14) Substitution |
| (6) New facts received | (15) Associative search |
| (7) Data line | (16) Set specification |
| (8) Facts | (17) Values of assertion functions |
| (9) One sentence of SL | |

The Logic Unit (LE) contains the definition of the variables, types and predicates needed to define the facts and on the basis of this it performs substitutions on the facts arriving from memory and generates the new assertion.

There can be any number of Logic Units since they can work in parallel, counting from the time of arrival of the facts. If several Logic Units are working in parallel then the several Logic Units also evaluate the assertion functions and predicates. For this reason we did not show this part in Diagram 4.

One of the two data lines forwards the original or the derived facts to the logic units and the other forwards them to memory.

In conclusion we would like to thank all those who have enriched this theme with their ideas. The number of tasks awaiting solution is very large and we have formulated only the first steps.

LITERATURE

1. P. Hajek and T. Havranek: "Mechanizing Hypothesis Formation," Springer Verlag Universites, 1978.
2. D. B. Lenat: "The Ubiquity of Discovery," Proc. of 5th IJCAI, Cambridge, 1977, pp 1093-1105.
3. Gabor David, Sandor Keresztely and Andras Sarkozy: "Concerning the Automation of Microprogramming," 1978 Programming Systems Conference of the Janes Neumann Society of Computer Technology, pp 127-137.
4. Gabor David, Sandor Keresztely, Ilona Losonezi and Andras Sarkozy: "Logic-Based Description of Microcomputers," MTA SZTAKI TANULMANYOK, No 91, 1978 pp 75-92.
5. Microprogram Synthesis, *ibid* (4), 187-206.
6. R. M. Burstall and J. A. Goguen: "Putting Theories Together to Make Specifications," *ibid* (2), pp 1045-1058.
7. Gabor David: "Structured Automatized Design of Microprograms," Large Scale Integration, Lawson et al eds., North-Holland, 1978.
8. Gabor David and Andras Sarkozy: "A Many-Sorted Logic Based Programming Language," Mathematical Logic in Computer Science, North-Holland Ser, Janes Bolyai Soc. (in preparation).
9. Gabor David: "Proving Correctness and Automatic Synthesis of Parallel Program," to be published, Algorithms, Software and Hardware of Parallel Computers, VEDA Publishing House (in Russian: Nauka, Novosibirsk).
10. Problemsolving Strategies, Winter School on Operating Systems, MTA SZTAKI TANULMANYOK, (being printed).

LOGIC MACHINES VERSUS NEUMANN-PRINCIPLE COMPUTERS IN TASK SOLVING

Budapest INFORMACIO ELEKTRONIKA in Hungarian No 1, 1980 pp 10-16

[Article by Tamas Gergely and Miklos Szots, scientific workers at the Computer Technology Research Institute: "Logic Machine Contra Neumann-Principle Machines in Task Solving"]

[Text] Recognizing the limitations of the traditional--Neumann-computer, the article recommends a logic machine which better simulates and is more suitable for task solving. The basic unit of the logic machine is able to generate logical conclusions and, proceeding through a rather long chain of conclusions, the solution desired can be obtained. One property of task solution is how the user is able to formulate his problem. He can express the information in higher order logic but conclusions can be drawn mechanically (that is with a logic machine), in a reassuring way only in logic lower than the second order. The authors recommend a method which reduced the "logic of formulation" to the "logic of solution" of the machine. They only illustrate this method for a formal discussion of it would require a profound mathematical apparatus and such a discussion would go beyond the frameworks of this article.

Introduction

The need for the swift and reliable performance of complex numeric calculations created the computer. This need defined the design principles of the computers still in use, the so-called Neumann principles:

--the computer is capable of executing a finite number of instructions, executing each of them in a finite time;

--the data for the computation can be accessed by means of the name (address) of the storage units;

--the programs describing the algorithms of the computations and the data for the computations can be represented in the same language.

Despite the fact that the computer is now regarded as the universal machine to aid intellectual work, these principles have not essentially changed.

Aiding intellectual work requires machines which can play a greater role in task solving than is the present practice of computer technology, that is the execution of an algorithm worked out for the solution. In what follows we will examine on the basis of what principles one could construct a computer which would be capable of producing a solution by starting from the formulation of the task, without the solution of an algorithm.

Task and Task Solution

If we are to talk about tasks and task solution we must distinguish between the task solving system and its environment, which, however, stand in a dynamic mutual relationship to one another.

Various influences from the environment affect the system--we call these stimuli--and the system responds to these, generally by somehow changing the environment. Let us presume that

--it is not a matter of indifference to the system what responses it gives to the stimuli; it has certain goals and motivations which encourage it to give "correct" responses;

--the system has a sort of "knowledge," some sort of description--generally not complete--of its environment, on the basis of which it can select the responses considered correct;

--it has some sort of inference mechanism with which it can produce responses not explicitly known.

If the above conditions are met then we call the system and the stimulus together a task situation. If we want to compare the awareness of the system and the character of the stimulus we can divide up the task situation into three groups:

--the response to be given to the stimulus or the algorithm which produces it exist in the knowledge of the system--we call this case an example;

--the response to be given to the stimulus can be determined from the knowledge available with the aid of the inference mechanism of the system--we then speak of a task;

--the system cannot respond to the stimulus on the basis of the knowledge available and the inference mechanism, a response can be given only by expanding the knowledge or the inference mechanism--then we speak of a problem.

In what follows we will deal with the case of tasks in the narrower sense although suitably supplemented the devices described by us could be applied to the solution of problems also.

In describing the task situation we used the most abstract possible approach. Making this more precise and conducting the study on this level of abstraction could lead to a general theory of intelligent systems. But we have a much more modest goal--to develop the basic principle for a machine which could play a substantive role in the solution of scientific and technical tasks. We could treat these tasks independent of the solving system; this is made possible by socially developed norms. In other words, the concept of a task can be interpreted independently of the solver for social agreement ensures suitably motivated task solvers. As a consequence of this we do not have to deal with motivation because the correctness of the solution can be judged at the social level and the motivation of the user is automatically realized at the level of the machine.

We should note that we are calling an adequate description of the motivating social norms a task.

Another circumstance of equal importance is that the above concept of task requires the use of some sort of language or languages. Knowledge and the task itself appear verbally, describable in some sort of language; indeed, the process of solution itself can be described as the sequential execution of linguistic (syntactic) rules.

This latter fact suggests that when automating task solution we must start from the language, as the tool of cognition, and the process of task solution can be treated as a textual transformation. Certain psychological research suggests the same thing.

For the most part psychological research distinguishes two aspects in the course of studying task solution. One is general, ensuring that the solution should be correct. According to this view this is ensured by adhering to the rules of formal logic. Thus formal logic is the general frame for task solution. The individual task solving activity of people can be considered a realization of this--and here enters the individual, psychological aspect. We know that formal logic does not provide a task solving algorithm but only those rules which must be adhered to. Thus if we speak of realization we must not think of the realization of an algorithm. Rather we should think of a realized algorithm (or algorithm schema or the ability to construct algorithms) which adheres to the rules of formal logic.

The fact that an algorithm is an important tool for task solution is also disclosed by psychological research which shows that when a task is being solved a series of operations made up of elementary operations (algorithms) are executed. The elementary operations are bound together by logical conditions; they decide the applicability of the appropriate operations or check its effect. When a person learns some type of task solution he first

fixes an algorithm for the solution of the several tasks and these then become more abstract and increasingly schematic. Finally, with the development of a stable solution only a logical frame remains, which is actually the definition of the solution. If the existing information must be used for a concrete task then a process takes place which is the reverse of the learning process and the logical frame is amplified into an algorithm.

Both the psychological and the systems theory approach show that the problem of task solution must be approached from the side of language and logic. So let us briefly review the definition and the most important properties of the concept of abstract language and logic.

Language, Logic and Task Solution

In accordance with our purposes we should regard language as a tool of cognition, disregarding the special question of communication. Because of the lack of space we will not prove the plausibility of the definitions given below. We deal with these more amply in reference 1.

We will call language a duality of syntax and semantics. Syntax, or F , we will identify with the set of expressions in the language (formulas or a series of correctly structured signs). We will require that this set be recursive; thus it can be decided whether or not a series of signs belongs to the language. By semantics we will mean the combination of the possible worlds (models) M and a function k (the function of meaning) which gives the meaning of the expression in the appropriate model for every model and every expression.

Thus we mean by language a triad (F, M, k) where

1. F is a recursive set consisting of a series of signs, and
2. k is an interpretative function for $M \times F$.

If the elements of M are mathematical objects then we speak of a mathematical language.

Frequently the values of k are a set of truth values (for example, $\{0, 1\}$). In this case we use the so-called validity relationship instead of k ; if $k(u, \varphi) = 1$ then $u \models \varphi$ in the contrary case $u \not\models \varphi$. (Here $u \in M, \varphi \in F$.) We read the expression $u \models \varphi$ to mean that φ is valid (true) in u , u satisfies φ , or u is a model of φ . If a formula is valid in every model we call it generally valid. We can expand the validity relationship to sets of formulas too; a formula set is valid in a model if every formula of it is valid in it.

The concept of semantic consequence is closely linked to the validity relationship. We say that the formula φ is a semantic consequence of the formula set Σ ($\Sigma \models \varphi$) if φ is valid in every model of Σ thus: $\Sigma \models \varphi \Leftrightarrow$ every $u \in M$, for which $u \models \Sigma$ is a model of φ , that is $u \models \varphi$.

We regard the elements of M as models of the world, which we characterize as elements of F . We call any formula set which has a model (thus, there is at least one model in which every formula of the formula set is valid) a theory. Thus we can characterize the world (the environment of the task solving system), or more precisely the information pertaining to the world, in two ways:

--as the class of models which satisfy our information; or

--as the set of formulas describing our information.

Naturally the model class alone does not determine the language because the syntax does not always precisely describe the models; thus not every model class can be unambiguously characterized with formulas.

But we can now see already that language is suitable for a representation of one important component necessary for task solution, knowledge. In what follows we will provide knowledge with a theory of a fixed language and we will call the elements thereof non-logical axioms. We will also presume that the task itself can be described in the language. The description of the task is an implicit definition; making this explicit is the solution of the task.

We showed in the preceding point that another condition for task solution is a text transformation mechanism which satisfies certain semantic criteria and is thus some sort of system of inference.

Let F be the syntax of the language, thus a set of well formed texts, which we can presume to be recursive. For any n the partial function leading from F^n to F is called the rule of deduction if it can be determined by the function, that is from any element of F^{n+1} , that for the first n elements the appropriate function gives the $n + 1$ element. A function leading from F^0 to F (thus a constant) gives a formula; and we call these logical axioms. The set which can be determined by the rules of deduction we call the Calculus.

The link between the calculus and semantics is missing from the above formal definitions. In principle this link could be of many types but the most significant property is so-called truth preservation. A rule of deduction is truth preserving if the formula obtained as a result is valid in the model class defined by the arguments. A calculus is truth preserving if every rule of deduction of it is so. A truth preserving calculus--as the definition implies--serves to produce the consequences of the theory. We should note that the goal in mind--task solution--requires a truth preserving calculus. So in what follows we will examine only a deductive calculus and thus we will regard only truth preserving rules of deduction as rules of deduction. In this case truth preservation is a necessary requirement of the calculus. In general we call this correctness. If we look at the partial functions produced by the sequential application of the rules of deduction then we can say of these simply that they will be truth preserving. The partial

functions composed at will from the rules of deduction we call the deductive scheme. Since every rule of deduction can be determined the rules can be made into algorithms and so we can regard the deductive schemes as algorithms too. We call the execution of the algorithm corresponding to the deductive scheme, starting from non-logical axioms, deduction and we call the result the deduced formula.

We call a formula deducible from an axiom set Ax by the calculus K ($Ax \vdash_K \varphi$), if one can build up from the rules of the calculus a deductive scheme which gives the formula φ as a result for some arguments within Ax . Depending on the relationship of semantic consequence and deductibility to one another we can specify complete calculuses. We call a calculus complete with which we can deduce all the consequences of every non-logical axiom system Ax :

$$Ax \models \varphi \Leftrightarrow Ax \vdash_K \varphi.$$

In this case the concept of semantic consequence and deductibility are by extension the same relationship.

We call the pair (LANGUAGE, CALCULUS) logic. The link between logic and task solution can be seen already from the above. We need only prove that the solution of a task is equivalent to the deduction of a formula. If our logic has a complete calculus this means that every task which can be formulated in the language can be solved. Now let us take another step and discuss the method of solution.

Let us look at the set of all formulas which can be deduced from the given non-logical axioms and the rules of deduction interpreted as operations on them. We call these taken together the search field. If the task can be solved the formula representing the solution figures in the search field. Thus we can use for task solution a procedure which gradually produces the elements of the search field, with the aid of the rules of deduction, and checks each of them to see if it is the solution of the task. The most important characteristic of this procedure is in what order it generates the elements of the search field. The factor which determines the order we call the search strategy. The simplest formulation of the search strategy is a partial ordering interpreted on the elements of the search field. The procedure always produces the minimal number of elements which can be generated according to the ordering. The calculus and the search strategy together are already a constructive tool for generation and thus for solving the tasks. We should note that we call the triad (NON-LOGICAL AXIOMS, CALCULUS, SEARCH STRATEGY) the proof procedure.

We should call attention to the fact that the calculus and the search strategy together already give an algorithm. The general steps of this are:

1. generate the next element according to the search strategy; if there is none then the process terminates unsuccessfully;

2. check to see if the newly generated element gives a solution; if yes then the process terminates successfully; if no then return to step 1.

We should note that the above algorithm is not identical to an algorithm given by a deductive scheme belonging to the theorem to be deduced.

In the next point we will examine to what extent a concrete logic, classical logic of the first order, is suitable as a tool for task solution.

Classical Logics and Task Solution

The best known languages of mathematical logic are the so-called classical languages. For this reason we forego a detailed description of them. Their basic characteristic is that their model class is a class of fixed type structures. (By structure we mean the combination of a set, all functions interpreted on it and their relations.) The type of language determined what relations and how many functions of what argument it has.

Many languages with various expressive power--containing one another--belong to the class of structures as a model class. The three most important are:

--a null order language, which can speak only of basic set elements which can be called constants (This is essentially equivalent to the so-called judgment calculus);

--a first order language, which can speak of the elements in the basic set in general, with the aid of variables interpreted as objects; and

--a second order language, which can speak of functions and relations, with the aid of function and relation variables.

The languages listed here can be expanded by one another and their expressive power increases with the increase in order. The second order language is already suitable for an unambiguous description of an optional model; a first order language is not.

At the same time the power of the calculus belonging to the language decreases with the increase in order. A first order language has a complete calculus; this cannot be found for one of the second order. On the basis of the above it would appear that a first order language was most suitable for task solution. It is known that a first order theory is suitable for describing a sphere of knowledge and, according to the foregoing, the language has a complete calculus. We must examine what class of task can be described in a first order language.

Accepting a first order language, we can speak of three types of task:

--The only question is, is a statement true in every model of the theory; that is, is it a consequence of a formula of the theory or not.

--We are seeking objects which satisfy certain specifications in every model of the theory.

--We are seeking functions or relations which satisfy certain specifications in every model of the theory.

Let us stop with these three cases. The first is the most simple; any complete proof procedure is suitable for a solution for the task is simply to prove a formula. In the second case the specification can be formulated as an open formula: $\varphi(\vec{x})$, where the $\mathcal{U}$ satisfying the specification applies to elements ($\vec{\alpha}$) in the model:

$$\mathcal{U} \vdash \varphi(\vec{x}) [\vec{x}/\vec{\alpha}].$$

Here the task can be formulated in the form $\exists \vec{x} \varphi(\vec{x})$ with the aid of an existential quantor. Let us see what solving the task means. Since the task solution takes place at the level of the syntax of the language we cannot regard the proof of $\exists \vec{x} \varphi(\vec{x})$ as a solution. (This only proves the existence of a solution.) The solution of the task is to find those expressions which satisfy φ by substitution in the place of $\vec{x}$:

$$\mathcal{U} \vdash \varphi(\vec{x}) [\vec{x}_1/\tau_1, \dots, x_n/\tau_n].$$

This is possible if the proof procedure (the calculus) is constructive--if it actually produces an expression for all existentially quantified variables which satisfies the formula. It is true that there is a complete constructive calculus for a first order language. Thus the tools of theorem proof are suitable for the solution of tasks of this type.

We should note that the existence of a constructive calculus naturally requires a satisfiable condition from the non-logical axioms. (Namely that there be universal formulas.)

In the third case the specification of the task is a formula in which there is a function or variable of relation. If we tie these down with an existential quantor we have expressed the task, but we do this in a second order language which does not have a complete calculus. But the solution of the task is not the proof of the existence of an appropriate function (relation) but rather the expression which describes this, the construction of a formula out of the function and relation signs determined by the type of language. And it can be proven that a complete proof procedure does exist for this. We should note that this problem and the above thesis raise interesting logical questions (definition theory questions, the interpretation of Henkin models, questions embedded in a first order language and the relations of these). We deal with these questions in detail in reference 2.

We might close this point by noting that first order classical logic is suitable for the formulation of tasks and the proof procedures thereof are suitable for solving tasks. Thus, instead of speaking of a "task solving machine" we can speak of a machine representing first order logic.

In what follows we will briefly consider what we mean by a "logic machine" and we can then turn to a description of the machine which interests us, one realizing first order logic.

Mathematical Machines

We can call a machine mathematical if it performs mathematical operations. Since the operations are mathematical we can at least expect that they can be described formally. So there should be a language $N = (F_N, M_N, k_N)$ the elements of the syntax of which can be made to correspond to elements of the machine (or to states thereof) and to the operations which can be performed among them. In this regard we can say that the machine realizes a model of the language. When we examine, from the viewpoint of automation theory, what language the machine (regarded as an automat) represents we are actually examining the significance function " k_N ," that is, the extent to which the elements of the syntax of which language can be depicted in the machine.

In general a concrete machine does not precisely correspond to a model of the language involved. For example, the models of arithmetic are infinite and so cannot be realized perfectly. Realization carries certain limitations. So we regard a model (or the theoretical construction totally equivalent thereto) as an abstract machine. Since our purpose is a theoretical investigation we will mean by machine, in general, an abstract machine thus interpreted.

Let us show a Neumann-principle computer to show that the above train of thought is justified. We can ignore a description of sign and floating decimal without limiting the generality. In this case a digital computer is a representation of natural numbers (a standard model of arithmetic):

--the bit combinations corresponding to the words depict the numbers, and

--the instructions represent the operations or relations interpreted for the numbers.

An important question arises in connection even with this example. For a long time we have not been using a Neumann-principle computer as an arithmetic machine and this seems to contradict our argument thus far. But let us think--in the final analysis we code every non-arithmetic application in arithmetic. It is also true in general that if the language of a machine has enough expressive power so that another mathematical discipline can be coded with the aid of it then the machine, with suitable coding, can be regarded as a machine for the other discipline. Since everything can be coded in arithmetic the Neumann machine can be regarded as a universal machine.

According to our concept of a machine a machine gives information about a model and not about the appropriate language. For example, an arithmetic machine (which, for example, realizes a standard model) will not figure out the truths valid in a first order theory of arithmetic but rather those valid within the sphere of natural numbers. In general, this is our goal also. The fundamental properties of a language (for example, the infinite model class) are necessary only because the individual models cannot always be completely described. This is also the situation for the standard model of arithmetic. The fact that the abstract machine can be only partially realized corresponds to the undescribability of it. For example, in an arithmetic machine the determination of a value satisfying a first order formula is possible only within the frameworks provided by the machine (only limited quantors can be realized).

In contrast to the foregoing machine concept we call a machine a logic machine if it represents a logic. For example, if a machine:

- can depict first order expressions pertaining to arithmetic;
- has an axiom system of arithmetic at its disposal; or
- can execute the deductive rules of a first order calculus,

then it is a logic machine for arithmetic. Even without any further explanation it can be seen that there is an essential difference between the two machine concepts--the first only represents a model directly, in a schematic manner, while the latter represents a model class which can be described by a language, indirectly and in a syntactic manner. (In addition, we use the name logic machine only as a distinction; if the language describing the proofs of the logic is mathematical then the corresponding logic machine will be a mathematical machine too.)

Let us look at the problem of use. We want to solve some practical task with the aid of a machine. Then, in general, we must carry out dual coding:

- the knowledge characterizing the task must be represented in a mathematical discipline, and
- the above discipline must be coded in the language of the machine.

In selecting the mathematical discipline figuring in the above we can start from the task or from the machine. We can say that we need a machine in which the mathematical tools needed to solve the task can be easily coded or that we must use tools which can be easily coded in the machine.

In the foregoing we found the mathematical discipline or mathematical logic adequate to machine task solution; concretely, we found classical first order logic. We are not now dealing with the problem of how this can be coded in an arithmetic machine but rather with the problem of what sort of machine can accept this directly, without coding. That is, what must a first order classical logic machine be like.

Construction and Use of a Logic Machine

A logic machine represents a logic, at the level of syntax; it depicts the expressions of the language and can execute the rules of the calculus. Our goal is the execution of proofs (since we can reduce the task solution to this). But in describing the basic concepts of logic we see that a logic in itself is not sufficient to generate proofs; this also requires some sort of scheme which orders the several steps given with the rules. We need a search strategy. This is not built into the machine; it actually takes over the role of a program. So we need formulas which the machine can manipulate --non-logical axioms and the theorem to be proved. These take over the role of data.

Let us examine the situation of the one using the machine. Traditional programming work consists roughly of two steps which correspond to the dual coding mentioned above:

--one must decide (or at least understand) what it is that the program must define, and

--one must construct a program (algorithm) which carries out the activity clarified in the preceding point; it must define the "how" of the task solution.

In practice we usually regard only the second as the programming work. The description of the task does not come directly into the computer processing, but only indirectly via the process of programming. Most modern programming technology experiments are trying to restore the first to its worthy place.

In the case of the logic machine the task is formulated in the language of logic, in the form of non-logical axioms and the theorem to be proved. No further transformations need be performed on these; these will play the role of processing data unchanged. The "how" of the processing is given with the formulation of the "program" (the search strategy), which means a work of formulation independent of the former. It can be seen that:

--the "dual coding" disappears;

--the functions of "what" and "how" are sharply distinguished, the data carry the first and the program carries the second;

--providing the "what" function, that is the formulation of the task, receives a direct role in the machine processing (the "how" no longer reflects the "what");

--as a result of the foregoing the formulation of the "what" will be of greater significance than the "how."

We should note that we could use, in the place of a first order logical language, a language more convenient than this which stands close to the natural language.

We have said that the programs of a logic machine are the descriptions of search strategies. This is not a forced analogy because the elementary instructions are the rules of the calculus and the search strategy says when which of them are to be applied. We generally understand by program the organization of elementary instructions into some procedure. Still, the search strategy, as a program, significantly deviates from the accustomed concept of a program. In the first place program means for us primarily a deterministic program. How could we describe a deterministic search strategy?

We determine what steps must be executed on what formula: thus we write down a proof. This writing means fixing upon an already known proof course, and this is not what we want to do. (We might formulate this by saying that the machine code programs of a logic machine are "proof checkers".)

What is really of interest to us are the non-deterministic search strategies, as programs. This brings up the question of programming languages. How are we to write them down? How are we to formulate such a search strategy? The most general method of formulation is a partial ordering defined in deductive steps.

The adequate language for the formulation of this is again a language of logic--but naturally a different type of language than that in which we describe the data. This language must speak of proofs and of the formulas of syntax. So an equivalent of the Neumann principle is valid for the logic machine--the data and the program can be written in the same language.

The interpreter of the "programming language" is a theorem prover which works on the theorem proof which represents the task solution.

In what follows we will examine what sort of architecture the nature of the logic machine requires.

We start from the fact that its set of instructions consists of the rules of a calculus.

As an example let us look at a well known deductive rule, the so-called modus ponens:

$$\varphi, \varphi \rightarrow \psi \vdash \psi$$

That is, if there are two formulas the first of which is an implication and the second of which is the first member of the implication then the latter member of the implication can be deduced. Execution of the rule can take place in two steps:

--recognizing that the two formulas have the structure cited and thus the rule can be executed on them, and

--constructing the deduced formula.

It can be seen that execution of the second step can represent no problem if the first has already been taken. But recognizing the applicability of the rule is not at all a trivial question. This requires some sort of "pattern matching" mechanism. The functioning of this is primarily a storage problem.

In a traditional computer the units arriving for representation are natural numbers. The natural representation of these can take place with a finite series of digits--for example a series of zeroes and ones. In the case of a logic machine the expressions of the language, the so-called correctly structured series of signs, are the units to be stored. But--according to the foregoing--the storage must primarily make possible the mechanism of pattern matching, thus the recognition of the stored text. We know very well that tree representation is truly adequate for this. Naturally the trees can be coded in series and this is done even in contemporary computer technology practice. But the processing of texts thus coded will be complicated and so a large number of programming textbooks are full of various tree-manipulation algorithms. So one condition for the realization of a logic machine at the hardware level will be the development of a storage element which realizes direct storage of trees.

Another question arises in connection with storage--the question of access. It is a basic characteristic of Neumann type machines that their storage units do not each represent definite values; rather, any value falling between the value limits can be depicted in them. These units or "compartments" play the role of variables used to describe the course of the calculation. Working with them requires addressing the "compartments" and in the case of more complicated calculations much of the processing ensures access to the appropriate values. This is virtually a law if we are coding tasks on an arithmetic machine because the values (numbers) are themselves only codes and their place in memory (their location) carries the data connected with the task.

In the case of a logic machine the stored unit is an expression of the language (or a constituent element thereof). Here also we describe the operations performed on them with variables, variables which operate on the expressions of the language. (In order to distinguish them from the variable signs of the language we call these metavariables. An example can be seen in the description of the modus ponens, given earlier as an example of a deductive rule.)

The difference from the previous case now is that it is not this identification which carries essential information but rather the stored text itself

(which, according to the foregoing, is a tree). Reference by means of a metavariable is necessary only in order that the stored text should have some definite structure. Thus access to the stored text takes place by means of a pattern matching between the reference and the text, as has been mentioned above. Thus the fact that the text is stored in some "compartment" loses its significance. (Naturally we do not want to exclude direct access without pattern matching, but it will be quite secondary.) Thus the complex manipulations with addresses are left out and we can say that a memory organization based on some sort of associative principle is adequate for the logic machine.

We have not yet spoken of one important problem--efficiency. The various automatic theorem provers implemented thus far can be conceived of as simulations of a logic machine on a Neumann type computer and their efficiency has not been adequate for practical tasks. We should mention two reasons for this. One is the fact of simulation itself. Although they might appear to be logic machines to the user their functioning is burdened with coding into arithmetic and with an address organized memory. We hope that the realization of logic machine hardware (at least something approximating hardware) might bring a significant change.

The other reason for the slight efficiency is that the automatic theorem provers were implemented with a built in search strategy. So if we look at them more closely they can hardly be regarded even as a simulation of a logic machine. The primary reason for this is that the role of search strategies has not been described in detail even theoretically. So we have before us another line of research in addition to an approximate realization of the hardware--a study of search strategies as programs.

Bibliographical Notes

The ideas which have been presented are not radically new; they have been "in the air" for some time. We are only undertaking to think them through consistently and draw the consequences from them.

The idea that a logical representation is useful for task solution can be found in Green, among others.³ Another important step was R. Kowalski's idea of "logic, as a programming language."⁵ We call attention to the fact, however, that the description given by us differs from his. The position of the PROLOG systems realized on the basis of "logic, as a programming language" lie between the traditional view and our own.

P. Hayes has called attention to the fact that the search strategy (the "how") must be distinguished from the description of the task and that two different languages are needed.⁴ He was also the one who brought up the description of the search strategy in the language of logic.

We owe our thanks to Gabor David, an exchange of ideas with whom initiated that train of thought the results of which we have summarized in our article.

He suggested the conception of the logic machine thought through here--a calculus realized at the hardware level as a set of instructions, a logical theory stored as data and a search strategy as a program.

LITERATURE

1. T. Gergely, and A. Szabolcsi: "How To Do Things with Model Theoretical Semantics," CL AND CL, Vol. XIII, 1979.
2. T. Gergely and M. Szots: "Logical Machines and Their Theoretical Bases," in preparation.
3. C. Green: "Application of Theorem Proving to Problem Solving," Proceedings of the IJCAI, Washington, 1969.
4. P. J. Hayes: "Computation and Deduction," Proceedings of the Second MFCS Symposium, Springer Verlag, 1973.
5. R. A. Kowalski: "Predicate Logic as Programming Language," Proceedings of the IFIP, North-Holland, 1974.

8984

CS0: 2502

SATELLITE TRACKING STATION, LASER RANGE-FINDER DESCRIBED

Warsaw SKRZYDLATA POLSKA in Polish No 47, 25 Nov 79 pp 3-4

[Article by Pawel Fisztein: "Satellites Over Borowiec"]

[Text] I have known for a long time that watches can be set with great precision in Borowiec. Also, it has been no secret that in Borowiec they measure distances to the artificial satellites flying over Poland. But it is one thing to know and another to see all that equipment in place. Hence, the subject of this installment of the Poland in Space series is Borowiec where for a visit with the kind permission of Professor Stanislaw Grzedzielski, director of the Polish Academy of Sciences Space Research Center (CBK). Borowiec belongs to the CBK as the Astronomical Latitude Observatory, one of three scientific centers forming the Planetary Geodesy Institute.

How to find your way to Borowiec? In the CBK in Warsaw I was advised to go west and to stay as close as possible to the 52nd parallel (an exact number of degrees and seconds was given), the same which passes through Irkutsk...I took the advice and I can boast that I found the place without a hitch.

Seriously speaking, the CBK station, Borowiec is located near Poznan, approximately on the 20th kilometer on the road to Koscian, in a secluded forested area. Right next to the woods, there is a sparkling clean one story small building with red tile roof. There is a porch with small columns, just like an old fashioned Polish country squire manor. I would have enjoyed coming here in a chaise pulled by a pair of white horses. I am visiting the observatory manager, Doctor Janusz Moczko.

I received quite a lot of information. Some of the data was of historical nature. The observatory was established in 1957. Its origins are in the accords between the Polish Academy of Sciences and the USSR Academy of Sciences. These accords dealt with the cooperation between the scientists of both countries in the matters of analysis and interpretation of the movement of the terrestrial pole and the changes of latitudes and time. It is known that the earth rotates around its axis and the axis does not hold steady its position in the earth mass. For this kind of research, two points

were selected, located on the same parallel and separated by about 90 degrees longitude. One of the geographic points, if that is the proper term, was the observatory in Irkutsk near Ozero Baykal and the other, thanks to the late Professor Witkowski of the Institute of Astronomy of the University of Poznan and thanks to the support of the funds provided by the Polish Academy of Sciences, was built right here in Borowiec. This location was selected because it is situated precisely on the required geographical coordinates. The work was started even before the first artificial satellite was put in orbit. The observations began in 1957. Doctor Moczko works here since 1958. He was born in Leszno, the gliding capital of Poland, of which he is proud. He has behind him 27 years of working. Since 1953 he worked for the Polish Academy of Sciences. He graduated from the University of Poznan, majoring in astronomy. He is one of the co-founders of the Borowiec observatory. He mentions the first visual observations of satellites. When the Interkosmos program began, says Doctor Moczko, then our a bit amateurish work became serious and professional. We got an identification number for our station and the observations coming from here have been always considered very important and exact. Ten years later, the CBK was established and we of the Institute of Astronomy of the University of Poznan were transferred via the Institute of Geophysics to the Institute of Planetary Geodesy which is managed by Assistant Professor Doctor Janusz Zielinski.

For the time being, I will put aside the history of the observatory as it is in effect a separate chapter, and I will move on to the current matters. The observatory has suitable equipment at its disposal. Among the equipment there are the special camera for photographing artificial space objects and the famous laser range finder. The camera is a Zeiss camera purchased four years ago. It is an early model suffering from certain troublesome operational inconveniences. Plainly speaking, it does not work without breakdowns. Nevertheless, it is used successfully to photograph the satellites passing over Poland. This camera is the SBG model by Carl Zeiss of Jena (Satellitenbeobachtungsgeraet) and it is a large item weighing 3,700 kg and 1.8 m tall. It permits to take pictures only at night when a satellite is visible against the starry sky. One cassette contains eight photosensitive glass plates (about 27 DIN), 9 x 12 cm size. During the flight of a satellite, about 16 pictures can be taken thanks to the automatic changing of the cassette. How precisely the distance to the satellite can be measured? The precision is about 10 to 20 m. Perhaps I should give a few more items of the data. The diameter of the receiving mirror is 50 cm. The camera uses an optical system based on the Schmidt telescope and uses two telescopes to search for the satellite and to track its movement. The camera can turn around its own axis (azimuth movement) and also it can move along the equator.

While taking pictures of the satellites, the camera can move in unison with the movement of the satellite. In this case, we obtain the picture of the satellite and of the stars in the form of dots. This is so thanks to the fact that while taking the picture, the photosensitive plate executes the reverse movement with the speed equal to that of stars. In the case in which

the camera moves in the same direction with the satellite and the photosensitive plate remains stationary, we will obtain the picture of the satellite as a light dot and the stars will be portrayed as dashes rather than dots.

Which satellites can we photograph? Well visible objects can be photographed which are in the field of vision of the camera. First of all, the special satellites are photographed: the geodesic satellites, the Geos-A and the Geos-C and, additionally, the satellites which do not have laser reflectors and, consequently, cannot be observed by the laser range finder. The latter are: Midas-4, Midas-7, Explorer-19 and Explorer-39. Of course, the most important are the two Geos satellites because they are observed simultaneously by the camera and the laser range finder.

Photographing is not the end of the work. It can be said that it is the beginning of the proper work. And it should be added, very tedious work. We have to try to find a barely visible light dot which like finding a very thin needle among thousands of dots and we have to try to determine the coordinates of the found dot--the satellite. First one has to find some well known stars in the given segment of the sky. And only then, on the basis of the precisely known positions of those stars, it is possible to determine the position of a given artificial satellite and its trajectory. This work involves the eyes of specialists, automatic equipment and computers. All observations made here are recorded. The control clock gives the time when the picture was taken. The results of the observations together with the plates are catalogued in a special library. All the observations are required for the geodesy. Naturally, the results are exchanged with other centers in the world.

In one of the rooms of the main building works Maria Bany, a laboratory technician. Located there is a piece of automatic equipment, so called comparator which serves to match photographic plates with pictures of stars and artificial satellites. This instrument makes it possible to determine the position of a satellite in relation to the stars. I check what can be seen in the viewfinder, an eyepiece magnifier. Against the black background of the film, only white dots and the coordinate net can be seen. The results are recorded by hand or automatically then sent to the University of Poznan where they are entrusted to a properly programed computer.

The famous laser range finder is also located in Borowiec. Why do I call it famous? Because it is stamped No. 1 and is the joint product of the states participating in the Intercosmos program: Czechoslovak Socialist Republic, German Democratic Republic, Poland, Hungarian People's Republic and the USSR. Dr. Moczko confesses that the laser range finder has been received but at that time we have not had a proper facility for housing it. We have constructed a makeshift facility but, as often is the case, the makeshift things last the longest. The range finder, being the prototype, has several imperfections but it is an instrument which permits us to take many measurements needed by the Polish geodesy and science. This instrument which, years ago, we probably could not construct ourselves, served as a point of departure of sorts for our independent efforts.

Already, we are building a new range finder which will be totally made in Poland. Many teams and enterprises are involved. As the production coordinator serves the Space Research Center and personally Dr. Hieronim Hurnik of the University of Poznan. The new range finder has been developed along the pattern of Internkosmos-1. It will be fully automatic and computer controlled. The existing laser is water cooled. Consequently, it cannot be used at low temperatures because the current facility cannot be heated. The new instrument will be housed in an air conditioned facility, leaving only the optics outside the building.

It is appropriate to refer at this point to the Directive of the Central Committee of the Seventh Congress of the PZPR in which Article 75 talks to, among other things, an increase in the role of science and Article 21 emphasizes the achievements of Polish science obtained through the participation in the Interkosmos Program (the manned flight in which a Pole took part) and the launching of the Kopernik-500 satellite; the two undertakings in which the same teams of technicians and scientists participated that are now working in the Space Research Center of the Polish Academy of Sciences.

How the measurements taken by the laser range finder are used? Among other uses, they are needed for geodesy. Also, the satellites and the laser measurements can be used for the observation of the movement of the earth and the location of its poles. The earth is subject to many gravitational forces known for a long time and to many which are unknown. For about 100 years, the international earth pole movement service has been in operation. Also, many observatories on a common parallel have been established. For a long time there has been the international time service. Once upon the time, an astronomical unit for measuring the time was in force and in connection with it, it was necessary to follow closely the earth rotational movement. Now we have the atomic time standard. The astronomers use this standard for analyzing the earth rotational movements. Thus, the situation has been reversed.

Borowiec is one of fifty stations in the world which form the latitudinal network and the time service. Each observatory sends the results of its observations to the Paris office or to Japan when they concern the movement of the earth poles and in turn the observatory receives a summary of all observations on the globe.

Fine, but what do we need such precise time for? When we have the time with such precision, we can determine the longitude with equal precision. And vice versa, when we have the longitude, we can determine the time. What precision is required? One thousandth of the second will do. For the needs of civilization in general, the precision of one half of the minute may be sufficient. For the needs of telecommunication, this is not enough. Simply, there is no electronic telecommunication without the atomic standard. I must add that the standard for the telecommunication requirements is located in Warsaw. In Borowiec, they only check the Warsaw time.

How can the good, precise time be obtained in Borowiec? They do have here many very good clocks and watches, including a quartz one. But it turns out that this collection is not absolutely sufficient. An atomic clock is indispensable, one that can reproduce the standard of the second ($1/31,556,925.9747$ of the solar year) with the relative error of 10^{-10} . Only this is the precise time we are talking about. An observatory to rate recognition in the scientific world should have three atomic clocks. There are about 20 such observatories in the world. Doctor Moczko assured me that Borowiec will get an atomic clock soon. But until then they make do with indirect methods. There is a method worked out by specialists from the Czechoslovak Socialist Republic and the German Democratic Republic which permits to obtain the time on the atomic scale through television links. Thanks to this method, the time can be measured in Borowiec with the precision of one millionth of the second.

The "time manager" in the Borowiec observatory is Dr. Ireneusz Dominski, one of the co-founders of the center and the designer of many systems in the time laboratory. He is a graduate of the University of Poznan, he majored in astronomy but he is also a first class electronic engineer. He explained to me in detail the time service operations, he showed me the instruments including some unique items which cannot be found anywhere else in Poland. The doctor showed me the clocks and among them one showing the central European time and another the stellar time. The latter is approximately 4 minutes fast per day. During the year it advances full 24 hours. This is related to the rotation of stars (i.e. of the earth). It serves the needs of astronomy and geodesy but only for certain calculations. Nowadays, it is more convenient to use the universal time. I looked at the cabinet containing the quartz clock. My inspection was interrupted by Dr. Dominski who was nearly shouting: "This is nothing, come in two months and you will see an atomic clock."

The best clock, however, cannot help without the capability to compare its readings with a definitive international standard. Thanks to the television system, the time can be measured down to one hundred millionth of the second. But, according to the doctor, this is kept in reserve. Recklessly, I asked whether they could use the time signals broadcast by all radio stations in the world. I was crashed by the look from my charming host. Every such signal contains an error and is unacceptable to the time service. The doctor reminds of the voyage by the spaceships Soyuz and Apollo and the time problems it involved. Simply, it was necessary to bring an atomic clock from the US which permitted both partners to operate on one common time. It had to be brought over, not transmitted via voice radio as this would not be precise enough for the space technology and would not provide sufficient precision for launchings, dockings in the terrestrial orbit and the execution of planned maneuvers. For instance, in Borowiec the television signals are used but the secret of success of this method lies in the fact that the whole measuring route has been calibrated, i.e., checked with a mobile atomic clock.

I am standing in front of an electronic counter showing millionth parts of the second. The doctor presses the right switch. Now we have the radio in the Federal Republic of Germany. Now we are measuring the Warsaw "long wave," i.e. the Radio Center, and we know what frequency stability it has. There are more than a dozen switches. The results can be read out immediately on the instrument screen. Constant monitoring is needed not only by those who perform the measurements. The results are periodically transmitted to, among others, the Communications Institute with which there is close cooperation in this field.

Now, it is high time to see the laser range finder. Dr. Stanislaw Schillak takes care of the laser. He guides me to separate building with the domed roof. The building is highly practical. We switch on an electric motor and in a moment the two halves of the dome open up and the building looks like a hangar. Inside there is the Interkosmos-1 range finder. It looks a bit like a photo camera and a bit like a search light or an astronomical telescope. The installation has a maze of cables and subsystems. To the side, in a small camping trailer, there is the control center, the range finder's brain.

Before I looked at the range finder, I had been told that in addition to serving the Interkosmos Program, our laser is known around the world. This is so because we cooperate with the US and the observations of artificial satellites (mostly American) made in Poland are an important supplement to the worldwide geodesic survey network. Borowiec cooperates with one observation point in Spain and one in Greece. Other Interkosmos type range finders are also located in Bolivia, Egypt (Helwan), India and Cuba. In the USSR there are two stations using range finders of this type.

This range finder was transferred to Poland from the observatory in Ondrejov in the Czechoslovak Socialist Republic in 1975. We are one of twenty countries having at their disposal this kind of satellite survey technique. The laser was built through the efforts of the Socialist countries in the Czechoslovak Socialist Republic. This is a ruby laser having the energy of 1.5 Joule. The width of the pulse is 25 nanoseconds (1 nanosecond = 10^{-9} sec). This is the duration of the pulse. The frequency of the pulse, the "light shot," is 10 seconds. This means that every 10 seconds there is an automatic laser "shot." The light exists through the lens visible in the picture. The lens reduces the divergence of the light beam to 0.8 milliradian. In practice, this means that at the distance of 1,000 km the width of the beam is 800 m. The tracking telescope is mounted on top of the instrument. This device is not automated and is intended exclusively for visual observation because satellites are easily visible and do not remain in the shadow of the earth. Of course, the observations are possible thanks to the receipt of exact coordinates of all satellites flying over Poland: every week special bulletins are received about artificial objects and their positions.

Which satellites do we use? Exclusively those having isometric prisms-reflectors. Thus, we are using the Geos-A and Geos-C satellites. Until recently, the Geos-B was helpful but it lost its properties because, unexpectedly, it started spinning on its own axis. Additionally, we are using the Interkosmos-17 satellite which is rather difficult to observe because it moves at a great speed as it is in a relatively low orbit around the earth. It is equipped with the prisms-reflectors made in the Czechoslovak Socialist Republic. The range of the range finder is 3,000 km. The precision of the range measurement is up to 50 cm.

The calculating routine is automated. The measurement control equipment is housed separately. We send a laser beam to the satellite. The beam is reflected by the prisms-reflectors on board and returns to the laser receiver. Knowing the time in which the beam (pulse, to be precise) covered the distance from the earth to the satellite and back to the earth, the distance to the satellite is calculated. Of course, the value for the speed of light is input to the computer with the corrections for the atmosphere and the vacuum. Thus, all the calculations are entrusted to the digital computer. I am simplifying the problem because before the measurements are sent to Poznan the electronic counters located here measure everything and the time at which the measurement was taken is registered. Special "laser's clock" prints coded information on paper. The complete result of the observation is obtained from the computer in readable form. The results show the data with the precision of 1 cm and also show the differences between the calculated, the expected and the measured distances. Here also, the precise time plays an important role. One could write a lot about lasers and range finders but I am terminating my visit to Borowiec, an installation of the Space Research Center of the Polish Academy of Sciences, which lasted 380 minutes and 11.234789 seconds.

Fig. 1. Dr. Janusz Moczko, the manager of the Space Research Center Observatory in Borowiec. P 3.

Fig. 2. The SBG camera used for photographing artificial satellites. P 3.

Fig. 3. Dr. Stanislaw Schillak by the Interkosmos-1 laser range finder. P 3.

Fig. 4. Dr. Ireneusz Dominski in the time laboratory. Also visible is the counter and two clocks. P 4.

Fig. 5. The laser range finder facility. P 4.

8801

CSO: 2602

END

SELECTIVE LIST OF JPRS SERIAL REPORTS

EASTERN EUROPE SERIAL REPORTS

EAST EUROPE REPORT: Economic and Industrial Affairs
EAST EUROPE REPORT: Political, Sociological and Military Affairs
EAST EUROPE REPORT: Scientific Affairs

WORLDWIDE SERIAL REPORTS

WORLDWIDE REPORT: Environmental Quality
WORLDWIDE REPORT: Epidemiology
WORLDWIDE REPORT: Law of the Sea
WORLDWIDE REPORT: Nuclear Development and Proliferation
WORLDWIDE REPORT: Telecommunications Policy, Research and Development

END OF

FICHE

DATE FILMED

APRIL-3-80

~~SS~~
D.D.